

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

**Нижегородский государственный университет  
им. Н.И.Лобачевского**

**Радиофизический факультет**

**Кафедра математики**

**ПРАКТИКУМ на ЭВМ: СБОРНИК ЗАДАНИЙ**

**Учебное пособие  
для студентов радиофизического факультета ННГУ**

**Издание второе, дополненное и исправленное**

**Нижний Новгород, 2010**

### **УДК 681.3**

Практикум на ЭВМ: Сборник заданий : Учебное пособие для студентов радиофизического факультета ННГУ. - 2-е изд., дополненное и исправленное / Сост. В.А.Савин. - Нижний Новгород: ННГУ, 2010. - 124 с.

В предыдущее издание 1996г. были включены варианты практических заданий, используемых в процессе преподавания общего курса "Вычислительные машины и программирование" на радиофизическом факультете ННГУ уже на протяжении длительного периода времени. Ранее в разработке этих заданий непосредственное участие принимали преподаватели кафедры математики В.А.Гусев, В.Б.Орлов, В.А.Савин, А.Ф.Шапкин. Проверенная временем методическая эффективность от использования комплекса таким образом подобранных заданий заставила свести их в единый сборник, позволяющий составить целостное представление о целях и задачах курса и заложить основы для дальнейшего изучения методов вычислений и математического моделирования на ЭВМ. Вместе с тем практика применения сборника выявила ряд вопросов, требующих постоянных пояснений для обучаемых. Это заставило коренным образом переработать ряд приводимых в сборнике примеров, снабдив их необходимыми комментариями. В ряде случаев предложены к выполнению упрощенные варианты заданий.

**Составитель: В.А.Савин**

**Рецензенты: В.Н.Белых, В.А.Вдовин**

Нижегородский государственный университет  
им. Н.И.Лобачевского, 2010

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b> .....	<b>4</b>
<b>ЗАДАНИЕ 1</b> .....	<b>6</b>
<b>ЗАДАНИЕ 2</b> .....	<b>12</b>
<b>ЗАДАНИЕ 3</b> .....	<b>20</b>
<b>ЗАДАНИЕ 4</b> .....	<b>25</b>
<b>ЗАДАНИЕ 5</b> .....	<b>41</b>
<b>ЗАДАНИЕ 6</b> .....	<b>60</b>
<b>ЗАДАНИЕ 7.1</b> .....	<b>73</b>
<b>ЗАДАНИЕ 7.2</b> .....	<b>87</b>
<b>ЗАДАНИЕ 8</b> .....	<b>106</b>
<b>ЗАДАНИЕ 9</b> .....	<b>119</b>
<b>ЛИТЕРАТУРА</b> .....	<b>123</b>

## ВВЕДЕНИЕ

Курс вычислительных машин и программирования является базовым курсом в системе математической подготовки на радиофизическом факультете ННГУ и ориентирован на освоение общих подходов к решению практических задач, связанных с использованием современных вычислительных средств и оргтехники.

Изучение методики постановки и решения вычислительных задач позволяет в конечном итоге добиваться от обучаемых логического обоснования при выборе средств достижения результата и проведения анализа этого результата.

Важнейшим этапом в достижении этой цели является изучение основ алгоритмизации и принципов использования алгоритмических языков программирования для постановки и решения прикладных задач на ЭВМ. Следуя выбранному направлению, задачи, вошедшие в сборник, расположены по мере усложнения их логической структуры. При этом каждый набор задач выбран для иллюстрации какой-либо характерной алгоритмической структуры.

Для каждого набора задач рассмотрен пример решения, включающий постановку задачи, обоснование подхода к ее решению и реализации программ на алгоритмических языках высокого уровня FORTRAN и C, иллюстрирующих возможности инструментальных средств решения практических задач.

При составлении вычислительно-машинных программ учитывались особенности способа задания числовых и символьных величин, присущих каждому из используемых языков программирования.

Всячески избегалось отступление от стандартных конструкций языка FORTRAN и от ANSI-стандарта языка C, что позволяет надеяться на достаточную мобильность предлагаемых программ, исключая высокую степень зависимости от архитектуры используемой ЭВМ и операционной системы, под управлением которой она работает.

Приведенные тексты программ на языках FORTRAN и C, ориентированы на использование практически любого современного компилятора этих языков программирования, работающих под управлением операционных систем UNIX и MS-DOS на ЭВМ класса IBM PC.

Для осознанного решения некоторых заданий требуются определенные знания материала курсов математического анализа, аналитической геометрии и высшей алгебры.

## ЗАДАНИЕ 1

На основе линейной алгоритмической структуры составить программу для ЭВМ, позволяющую вычислить результат арифметического выражения при заданных значениях переменных  $x$  и  $a$ .

### Варианты задания

$$1. y = \frac{\lg(1 + e^{|x|+1}) + \pi^e}{\sin(\frac{\pi+a}{e} + x) \cdot \operatorname{arctg}(1 + \sqrt{x} + x^2) + 14.8}, \quad \begin{matrix} x = 1.7 \\ a = 0.5 \end{matrix}$$

$$2. y = \frac{\sqrt[3]{e} \cdot \sin(x + \pi) - e^{-x^2}}{\operatorname{arctg}(1 + \sqrt[3]{x} + x^5) \cdot \ln(3.2 - |x - a|)}, \quad \begin{matrix} x = 0.5 \\ a = 2.2 \end{matrix}$$

$$3. y = \frac{(x^5 + x^7) \cdot \operatorname{arctg}(1 + \lg|x|) + |x - a| - 0.62}{11.5 + \sqrt{x^3} + \sin x \cdot \cos e}, \quad \begin{matrix} x = 0.98 \\ a = 3.1 \end{matrix}$$

$$4. y = \frac{\sqrt[3]{a + 1.7x^4} - e^{x\pi} \cdot \operatorname{arctg}(\sin(\frac{x+e}{a\pi}) + 1)}{\ln(a + e^{|x|}) + \lg(1 + e^{-x^2})}, \quad \begin{matrix} x = 0.9 \\ a = 5.1 \end{matrix}$$

$$5. y = \frac{\sin(1 + \sqrt{x} + x^3) + \sqrt[5]{e}}{\lg(1 + \ln(1 + e^{|x|})) + 4.2\pi^e - |11x - 4a|}, \quad \begin{matrix} x = 0.8 \\ a = 2.2 \end{matrix}$$

$$6. y = \frac{\sqrt[5]{x^3 + 1/x} \cdot (1 + \lg|x|)}{\operatorname{arctg}(e^{|ax^3|} - 0.59 \sin(x/a)) + \sqrt{a^e}}, \quad \begin{matrix} x = 1.2 \\ a = 0.5 \end{matrix}$$

$$7. y = \frac{e^{-x^2} + \sin(\pi\sqrt{a} + x^e) - \sqrt[3]{ax^7}}{\ln(a + x \cdot \lg(1 + e^{|x|})) + 3|a - x| - 2.4}, \quad x = 1.5, \quad a = 0.7$$

$$8. y = \frac{e^{xa} \cdot \sqrt[5]{x} + 7a^5 + 14.3(\sqrt{a} + \sin(x\pi))}{\operatorname{arctg}\left(\frac{a+\pi}{x^2}\right) - \lg(1 + a^e)}, \quad x = 1.4, \quad a = 1.2$$

$$9. y = \frac{\sqrt{e} \cdot \sin(a + x\pi) - \sqrt[3]{x^5 + a^7}}{\operatorname{arctg}(\cos(e\pi) - e^{-x}) + \lg(1 + |ax|)}, \quad x = 0.2, \quad a = 1.6$$

$$10. y = \frac{11.3 \sin(1 + \cos(a^2 + x^3)) - a^{e^2}}{\lg(1 + \operatorname{arctg}(1 + e^{|x|})) + 1.7\sqrt{x^7}}, \quad x = 1.3, \quad a = 1.6$$

$$11. y = \frac{\operatorname{arctg}(1 + x^2 - \sqrt[3]{e + \pi^2}) + 11.3e^{-|x|}}{\sin(ax + \sqrt{\pi}) - \ln(1 + e^{x^2})}, \quad x = 1.3, \quad a = 1.6$$

$$12. y = \frac{\sin(1 + e^{x\pi}) + \sqrt{x} + x^3 - \pi\sqrt{a}}{\ln(a^2 + e^{-x^2}) - \operatorname{arctg}(\cos(1 + |x|a^e))}, \quad x = 0.7, \quad a = 1.3$$

$$13. y = \frac{\ln(1 + e^{-x} + a^{|x|}) + 3.5\sqrt[3]{ea^4}}{\sin(a^e + x\pi) - \operatorname{arctg}(e^{-x^2} + \cos(e\pi))}, \quad x = 0.5, \quad a = 1.8$$

$$14. y = \frac{e^{-ax^2} \cdot \sin(x\pi + 0.7) + \lg(1 + e^{-|x|}) - 1}{0.56 + 14.7\sqrt{x^3} + |\sin(x/a)|}, \quad x = 0.8, \quad a = 1.1$$

$$15. y = \frac{\operatorname{arctg}\left(\frac{a+x^2}{\pi}\right) - 11.86 \sin(1 + \ln|a|)}{\lg(a^2 + e^{-x^2}) + 1.7\sqrt{x^7}}, \quad x = 1.4, \quad a = 0.95$$

$$16. y = \frac{\sin(a\sqrt{\pi}) + a^{e^x} - \sqrt[3]{x + \sqrt{x} + a^4}}{\ln(1 + e^{-|x|}) \cdot \operatorname{arctg}(1 + \pi/x^2)}, \quad x = 0.2, \quad a = 1.3$$

$$17. y = \frac{\operatorname{arctg}(1 + \lg|x|) - 1.7(\sqrt{a} + \sin(x\pi))}{\cos(e\pi) \cdot e^{-x} + \ln(x + x^7 + \sqrt{a^5})}, \quad x = 0.36, \quad a = 1.4$$

$$18. y = \frac{4.3 + \operatorname{arctg}\left(\frac{a+\pi}{x}\right) - \sqrt{x + 6a^5}}{\lg(1 + a^e) \cdot (1 + \ln(1 + a^{-e|x|}))}, \quad x = 1.2, \quad a = 0.3$$

$$19. y = \frac{\cos(x + \sqrt{x}) - \sin(ax^2 + \sqrt[3]{e} - 1/\pi)}{\ln(e^{-x^2} + 1.4 a^{-e}) \cdot \operatorname{arctg}(\sqrt[3]{x})}, \quad x = 1.6, \quad a = 1.5$$

$$20. y = \frac{e^{-ax^2} \cdot \lg(1 + \sin(\pi + x\sqrt{a})) - 1.3 a^e}{\ln(3.6 + \cos(\sqrt[3]{x} + x^4)) \cdot |a|^x}, \quad x = 0.4, \quad a = 1.3$$

$$21. y = \frac{\sqrt[5]{x^3 + x^2 + \sqrt{x}} + \sin(x\sqrt{a\pi}) - 1}{\operatorname{arctg}\left(\frac{a+\pi^e}{x^2}\right) - a\sqrt{e}}, \quad x = 1.5, \quad a = 0.35$$

$$22. y = \frac{\cos(a^2 + \pi x^5) + \operatorname{arctg}(1 + e^{-|x|})}{\ln(1 + a^e + \sqrt{\pi}) - \lg(|x| - 1.3)}, \quad x = 1.6, \quad a = 0.92$$



$$23. y = \frac{3x^2 + \sqrt[5]{x} - \operatorname{arctg}\left(\frac{e^\pi}{x^2+a^2}\right)}{\ln(1 + e^{-|x|}) \cdot \sin(a\pi + \cos(a/\pi))}, \quad x = 0.7, \quad a = 1.4$$

$$24. y = \frac{1.6x^2 \cdot \cos(e\pi) - e^{-ax} \cdot \ln(1 + |a|x^5)}{(\sqrt[3]{x} + \sqrt[3]{e}) \cdot \operatorname{arctg}(3.6 + \sin\sqrt{a})}, \quad x = 0.86, \quad a = 1.25$$

$$25. y = \frac{\sqrt[5]{e} + \sqrt[3]{a^7 + \pi|x|} - \ln(1 + e^{-x^2})}{\cos(a^2 + e/\pi) \cdot \operatorname{arctg}\left(\frac{1}{|x|+a}\right)}, \quad x = 0.3, \quad a = 1.76$$

$$26. y = \frac{1.3 \sin(a\sqrt{\pi}) + a^{\sqrt{e}} - |x| \cdot \ln(1 + e^{-ax^2})}{\operatorname{arctg}(e\pi) \cdot \lg(1 + x^5 + \sqrt[3]{x})}, \quad x = 0.7, \quad a = 1.6$$

$$27. y = \frac{\lg(1 + a^e) + e^{ax} - \cos(1 + |ax^5|)}{\sqrt{x^3 + \operatorname{arctg}\left(\frac{a+\pi}{e}\right)} \cdot \ln(\sqrt[5]{x} - x^4 + 1)}, \quad x = 0.95, \quad a = 1.3$$

$$28. y = \frac{\operatorname{arctg}\left(\cos\left(\frac{\pi+a}{x}\right) - \sin(1 + a^{ex})\right) - \sqrt[3]{e}}{\ln(1 + e^{-a|x|+x^2}) \cdot \cos(1 + |x|a^e)}, \quad x = 0.72, \quad a = 1.52$$

$$29. y = \frac{\lg(\sqrt{a} + \sin(x\pi)) + \cos(\sqrt[3]{x} + x^7)}{\ln(1 + a^e + \sqrt{\pi}) - |x| \cdot \lg(1 + \operatorname{arctg}\left(\frac{\sqrt{\pi}}{e-a}\right))}, \quad x = 0.2, \quad a = 1.2$$

$$30. y = \frac{e^{-ax^2} \cdot (1 + \sin(\pi + x\sqrt{a})) - \lg(1 + a^e)}{\operatorname{arctg}(\sqrt[3]{x} + 1.7\sqrt[7]{x}) + \ln(1 + e^{|x|})}, \quad x = 0.89, \quad a = 1.32$$

**Пример 1.** Дано арифметическое выражение

$$y = \frac{\sqrt[5]{x} \cdot \sin\left(\frac{\pi-x}{a}\right) + (x + 4/3)^3 - \pi\sqrt{a}}{\pi \cdot \lg(16.5 + e^{-a^2}) + \operatorname{arctg}(a^e - \cos(|x - 2|))}$$

и значения переменных  $x = 0.6$ ,  $a = 1.4$ , при которых требуется вычислить его результат. Линейная алгоритмическая структура предполагает последовательный порядок выполнения требуемых действий. При составлении программы для ЭВМ рекомендуется разбить числитель и знаменатель предложенного арифметического выражения на достаточно простые части. Это позволит избежать ошибок при записи слишком длинных выражений. При желании, однако, можно воспользоваться правилами записи строк продолжения.

```

C      ЗАДАНИЕ 1. Пример 1. Прогр. на языке FORTRAN.
      PROGRAM T1
      X = 0.6
      A = 1.4
      P = 3.14159265
      B = X**0.2 * SIN( (P-X) / A )
      B = B + (X+1.3333333)**3 - P*SQRT( A )
      C = P * ALOG10( 16.5 + EXP( -A*A ) )
      C = C + ATAN( A**EXP(1.) - COS( ABS( X-2. )))
      Y = B / C
      WRITE(*,*) 'X =',X, ' ; A =',A, ' ; Y =',Y
      STOP
      END

```

```

X = 6.000000E-01; A = 1.400000; Y = 8.768302E-01

```

```

/* ЗАДАНИЕ 1. Пример 1. Программа на языке C. */

#include <stdio.h>
#include <math.h>

int main( void ) {

double x = 0.6, a = 1.4, p = 3.141592653589793;
double b, c, y;

    b = pow( x, 0.2 ) * sin( (p-x) / a );
    b += pow( x+1.3333333333333333, 3.) - p*sqrt( a );
    c = p * log10( 16.5 + exp( -a*a ) );
    c += atan( pow( a,exp(1.) ) - cos( fabs( x-2.)) );
    y = b / c;

    printf( "x = %g; a = %g; y = %.16g\n", x,a,y );

    return 1;
}

x = 0.6; a = 1.4; y = 0.8768304153981453

```

## ЗАДАНИЕ 2

На основе разветвляющейся алгоритмической структуры составить программу для ЭВМ, позволяющую вычислить результат сложного арифметического выражения в зависимости от численных значений независимых переменных, которые определяют, в свою очередь, логические значения приведенных условий. Продемонстрировать работу программы для всех возможных случаев ветвления.

### Варианты задания

$$1. \quad y = \begin{cases} a \cdot \cos x, & ax \leq -1 \\ e^{-ax}, & -1 < ax \leq 2 \\ a \cdot \sin x, & ax > 2 \end{cases}$$

$$2. \quad y = \begin{cases} c \cdot \operatorname{arctg} x^2, & cx > 2 \\ 1/cx, & 0 < cx \leq 2 \\ c|x|, & cx \leq 0 \end{cases}$$

$$3. \quad y = \begin{cases} |x|/a^2, & ax^2 > 3 \\ \operatorname{tg}(ax), & 1 < ax^2 \leq 3 \\ e^x, & ax^2 \leq 1 \end{cases}$$

$$4. \quad y = \begin{cases} \sqrt[3]{|px|}, & px < e \\ \sin(p-x), & e \leq px \leq \pi \\ \operatorname{arctg}(p/x), & px > \pi \end{cases}$$

$$5. \quad y = \begin{cases} b^2 + x^2, & bx \leq \pi \\ x/|b|, & \pi < bx \leq 5 \\ e^{-bx^2}, & bx > 5 \end{cases}$$

$$6. \quad y = \begin{cases} |x| + |d|, & d/x > 1 \\ d/x, & -1 < d/x \leq 1 \\ \operatorname{tg}|dx|, & d/x \leq -1 \end{cases}$$

$$7. \quad y = \begin{cases} \lg|gx|, & \operatorname{tg}(gx) \leq 0.3 \\ \sin x, & 0.3 < \operatorname{tg}(gx) \leq 0.5 \\ g|x|, & \operatorname{tg}(gx) > 0.5 \end{cases}$$

$$8. \quad y = \begin{cases} x|a|, & ax \leq 3 \\ a/x, & 3 < ax \leq 5 \\ a \cdot e^x, & ax > 5 \end{cases}$$

$$9. \quad y = \begin{cases} \lg|x|, & a \cdot \ln|x| > 3 \\ ax, & 1 < a \cdot \ln|x| \leq 3 \\ e^{-ax}, & a \cdot \ln|x| \leq 1 \end{cases}$$

$$10. \quad y = \begin{cases} \cos x, & |ax| > 2 \\ |a - x|, & 1 < |ax| \leq 2 \\ e^{a-x}, & |ax| \leq 1 \end{cases}$$

$$11. \quad y = \begin{cases} cx^2, & cx \leq -1 \\ c \cdot \ln|x|, & -1 < cx \leq 1 \\ e^{-cx}, & cx > 1 \end{cases}$$

$$12. \quad y = \begin{cases} x^3 + p^3, & px^2 > 5 \\ |x| + |p|, & e < px^2 \leq 5 \\ x \cdot \cos p, & px^2 \leq e \end{cases}$$

$$13. \quad y = \begin{cases} z + e^x, & xz^2 \leq 1 \\ \operatorname{tg}(xz), & 1 < xz^2 < \pi \\ x/|z|, & xz^2 \geq \pi \end{cases}$$

$$14. \quad y = \begin{cases} |x/a|, & x \cdot \ln|a| > 2 \\ ax^3, & 1 < x \cdot \ln|a| \leq 2 \\ e^{-ax}, & x \cdot \ln|a| \leq 1 \end{cases}$$

$$15. \quad y = \begin{cases} b + x^2, & \sqrt[3]{xb} \leq 1 \\ b \cdot e^x, & 1 < \sqrt[3]{xb} \leq 3 \\ \sqrt{|bx|}, & \sqrt[3]{xb} > 3 \end{cases}$$

$$16. \quad y = \begin{cases} e^{a+x}, & ax > 1 \\ |a - x|, & -\pi \leq ax \leq 1 \\ \operatorname{tg}(ax), & ax < -\pi \end{cases}$$

$$17. \quad y = \begin{cases} e^{cx}, & c/x \leq 0.5 \\ c/|x|, & 0.5 < c/x \leq 1 \\ \operatorname{tg} x, & c/x > 1 \end{cases}$$

$$18. \quad y = \begin{cases} q^2 \sqrt{x}, & q - x \leq 2 \\ \sin(qx), & 2 < q - x \leq 4 \\ x \cdot \cos q, & q - x > 4 \end{cases}$$

$$19. \quad y = \begin{cases} bx^2, & \ln|bx| \leq e \\ b - x, & e < \ln|bx| \leq 4 \\ x \cdot e^{-b}, & \ln|bx| > 4 \end{cases}$$

$$20. \quad y = \begin{cases} e^p + e^x, & p^2 x^3 > 6 \\ x + p^3, & 1 < p^2 x^3 \leq 6 \\ \sin(px), & p^2 x^3 \leq 1 \end{cases}$$

$$21. \quad y = \begin{cases} px, & px \leq \pi \\ -p \cdot e^x, & \pi < px \leq 8 \\ \operatorname{tg}(x/\pi), & px > 8 \end{cases}$$

$$22. \quad y = \begin{cases} a \cdot \ln|x|, & ax^3 > 5 \\ a^3 - x^3, & 1 < ax^3 \leq 5 \\ \operatorname{tg}|ax|, & ax^3 \leq 1 \end{cases}$$

$$23. \quad y = \begin{cases} (e + b)^x, & e^{bx} > 5 \\ \pi^e - x, & 1 < e^{bx} \leq 5 \\ \operatorname{tg}(bx), & e^{bx} \leq 1 \end{cases}$$

$$24. \quad y = \begin{cases} \cos(px), & xp^2 > 2 \\ e^{-\pi x}, & 1 < xp^2 \leq 2 \\ p \cdot \sin x, & xp^2 \leq 1 \end{cases}$$

$$25. \quad y = \begin{cases} \operatorname{tg} x^2, & a/x \leq 1 \\ a|x|, & 1 < a/x \leq 3 \\ e^x - a, & a/x > 3 \end{cases}$$

$$26. \quad y = \begin{cases} \lg|x + b|, & |xb| \leq 3 \\ b \cdot \ln|x|, & 3 < |xb| \leq 5 \\ \operatorname{tg} x - e, & |xb| > 5 \end{cases}$$

$$27. \quad y = \begin{cases} c^4 + x, & |cx| \leq 1 \\ \sin c, & 1 < |cx| \leq \pi \\ |x| \cdot e^c, & |cx| > \pi \end{cases}$$

$$28. \quad y = \begin{cases} a \cdot \cos x, & xa^3 \leq -3 \\ \operatorname{tg}(ax), & -3 < xa^3 \leq 4 \\ a^3 + x^5, & xa^3 > 4 \end{cases}$$

$$29. \quad y = \begin{cases} \pi + d, & \sqrt[3]{dx} \leq 2 \\ \operatorname{lg}|x|, & 2 < \sqrt[3]{dx} \leq 6 \\ x \cdot \operatorname{tg} d, & \sqrt[3]{dx} > 6 \end{cases}$$

$$30. \quad y = \begin{cases} \operatorname{arctg} x, & bx^2 > 3 \\ x \cdot \ln|b|, & 1 < bx^2 \leq 3 \\ |b - x|, & bx^2 \leq 1 \end{cases}$$

**Пример 2.** Дано сложное арифметическое выражение

$$y = \begin{cases} e^z, & zt \leq -2 \\ \sin t, & -2 < zt \leq 3 \\ \sqrt{zt}, & zt > 3 \end{cases}$$

Введенные численные значения независимых переменных  $z$  и  $t$  определяют разные логические результаты приведенных условий. Все возможные случаи ветвления могут быть учтены оценкой значений только двух выражений отношения. Третий случай получается автоматически исключением из уже рассмотренных.



Разветвляющаяся алгоритмическая структура предполагает выполнение действий, предусмотренных прохождением только по одной из возможных ветвей.

```

C      ЗАДАНИЕ 2. Пример 2. Прогр. на языке FORTRAN.
      PROGRAM T2
22     WRITE(*,*)
      WRITE(*,*) 'INPUT Z, T:'
      READ(*,*) Z, T
      D = Z * T
C-----
      IF(D .GT. 3.) GOTO 66 ! IF(D .LE. -2.) THEN
      IF(D.GT.-2.) GOTO 44 ! K = 1
      K = 1 ! Y = EXP( Z )
      Y = EXP( Z ) ! ELSE IF(D.LE.3.)THEN
      GOTO 88 ! K = 2
44     K = 2 ! Y = SIN( T )
      Y = SIN( T ) ! ELSE
      GOTO 88 ! K = 3
66     K = 3 ! Y = SQRT( D )
      Y = SQRT( D ) ! END IF
C-----
88     WRITE(*,*) 'Z =', Z, '; T =', T, '; D =', D
      WRITE(*,*) K, '-я ветвь; Y =', Y
      GOTO 22
      END

```

```
INPUT Z, T:
1. -3.
Z = 1.000000;    T = -3.000000;    D = -3.000000
           1-я ветвь;           Y = 2.718282
```

```
INPUT Z, T:
0. 0.
Z = 0.000E+00;    T = 0.0000E+00;    D = 0.0000E+00
           2-я ветвь;           Y = 0.000000E+00
```

```
INPUT Z, T:
3. 3.
Z = 3.000000;    T = 3.000000;    D = 9.000000
           3-я ветвь;           Y = 3.000000
```

```
/* ЗАДАНИЕ 2. Пример 2. Программа на языке C. */
```

```
#include <stdio.h>
#include <math.h>

int main( void ) {

int k;
double z, t, d, y;

l: printf( " \n " );
   printf( "input z, t : " );
   scanf ( "\f %lg %lg", &z, &t );

   d = z * t;
```

```

if( d <= 3. ) {
    if( d <= -2. ) { k = 1; y = exp( z ); }
    else { k = 2; y = sin( t ); }
}
else { k = 3; y = sqrt( d ); }
printf( "z = %g; t = %g; d = %g \n", z,t,d );
printf( "    %d-я ветвь; y = %.16g \n", k,y );
goto l;
}

```

```

input z, t : 1. -3.
z = 1; t = -3; d = -3
    1-я ветвь; y = 2.718281828459045

```

```

input z, t : 0. 0.
z = 0; t = 0; d = 0
    2-я ветвь; y = 0

```

```

input z, t : 3. 3.
z = 3; t = 3; d = 9
    3-я ветвь; y = 3

```

### ЗАДАНИЕ 3

На основе циклической алгоритмической структуры с явной организацией управлением переменными цикла составить программу для ЭВМ, позволяющую вычислять результат сложного арифметического выражения в соответствии с вариантами ЗАДАНИЯ 2. Подобрать параметры циклов (начальные значения, шаги изменения, конечные значения) для варьируемых переменных таким образом, что в процессе полного перебора возможных сочетаний этих переменных достигались все возможные случаи ветвления при расчете исходного выражения. Ограничить объем выводимой информации размером экрана видеотерминала.

**Пример 3.** В примере 2 к ЗАДАНИЮ 2 дано сложное арифметическое выражение. Часть программы, реализующей разветвляющуюся алгоритмическую структуру расчета этого выражения, может оставаться неизменной и составлять тело циклической структуры при выполнении обсуждаемого задания. Вместе с тем, определяющие результат численные значения независимых переменных  $z$  и  $t$ , уже должны задаваться непосредственно двумя циклами. Вложенность того или иного из них может быть задана по желанию пользователя. Подбор вводимых начальных значений  $z_1$  и  $t_1$ , шагов изменения  $z_s$  и  $t_s$ , конечных значений  $z_u$  и  $t_u$  для варьируемых переменных  $z$  и  $t$  позволяет добиваться искомого числа выполнений тела цикла.

```
C      ЗАДАНИЕ 3. Пример 3. Progr. на языке FORTRAN.  
      PROGRAM T3  
5     WRITE(*,*) 'INPUT ZL, ZU, ZS, TL, TU, TS: '  
      READ(*,*) ZL, ZU, ZS, TL, TU, TS
```

```

C
  Z = ZL
15  T = TL      ! Начало цикла по переменной Z
35  D = Z * T  ! Начало цикла по переменной T
C
      IF( D .GT. 3. )      GOTO 75
      IF( D .GT. -2. )    GOTO 55
      K = 1
      Y = EXP( Z )
      GOTO 95
55  K = 2
      Y = SIN( T )
      GOTO 95
75  K = 3
      Y = SQRT( D )
95  WRITE(*,*)
      WRITE(*,*) 'Z =',Z, ' ; T =',T, ' ; D =',D
      WRITE(*,*) K, '-я ветвь;          Y =',Y
C
      T = T + TS
      IF( T .LE. TU )    GOTO 35 ! Конец цикла по T
C
      Z = Z + ZS
      IF( Z .LE. ZU )    GOTO 15 ! Конец цикла по Z
      GOTO 5
      END

```

```
INPUT ZL, ZU, ZS, TL, TU, TS:
```

```
0. 3. 3. -3. 3. 3.
```

```
Z = 0.00000E+00; T = -3.00000; D = 0.00000E+00  
2-я ветвь; Y = -1.411200E-01
```

```
Z = 0.00000E+00; T = 0.00000E+00; D = 0.00000E+00  
2-я ветвь; Y = 0.000000E+00
```

```
Z = 0.00000E+00; T = 3.00000; D = 0.00000E+00  
2-я ветвь; Y = 1.411200E-01
```

```
Z = 3.00000; T = -3.00000; D = -9.00000  
1-я ветвь; Y = 20.085540
```

```
Z = 3.00000; T = 0.00000E+00; D = 0.00000E+00  
2-я ветвь; Y = 0.000000E+00
```

```
Z = 3.00000; T = 3.00000; D = 9.00000  
3-я ветвь; Y = 3.000000
```

```
/* ЗАДАНИЕ 3. Пример 3. Программа на языке C. */
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main( void ) {
```

```
int k;
```

```
double z, zl, zu, zs, t, tl, tu, ts, d, y;
```

```

1: printf( "input  z1, zu, zs, t1, tu, ts : " );
   scanf( "\f %lg %lg %lg %lg %lg %lg",
          &z1, &zu, &zs, &t1, &tu, &ts );

z = z1;
do { /* Начало цикла по переменной z */

    t = t1;
    do { /* Начало цикла по переменной t */

        d = z * t;
        if( d <= 3. ) {
            if( d <= -2. ) { k = 1; y = exp( z ); }
            else { k = 2; y = sin( t ); }
        }
        else { k = 3; y = sqrt( d ); }
        printf( "\n" );
        printf("z = %g; t = %g; d = %g\n", z,t,d);
        printf(" %d-я ветвь; y = %.16g\n",k,y);

        t += ts;
    } while( t <= tu ); /* Конец цикла по t */

    z += zs;
} while( z <= zu ); /* Конец цикла по z */
goto 1;
}

```

```
input z1, zu, zs, t1, tu, ts : 0. 3. 3. -3. 3. 3.
```

```
z = 0; t = -3; d = -0  
2-я ветвь; y = -0.1411200080598672
```

```
z = 0; t = 0; d = 0  
2-я ветвь; y = 0
```

```
z = 0; t = 3; d = 0  
2-я ветвь; y = 0.1411200080598672
```

```
z = 3; t = -3; d = -9  
1-я ветвь; y = 20.08553692318767
```

```
z = 3; t = 0; d = 0  
2-я ветвь; y = 0
```

```
z = 3; t = 3; d = 9  
3-я ветвь; y = 3
```



## ЗАДАНИЕ 4

Используя различные способы организации циклических алгоритмических структур, составить программу для ЭВМ, позволяющую вычислять в задаваемом диапазоне изменения независимой переменной (начальное значение, шаг изменения, конечное значение) функций  $\exp()$ ,  $\lg()$ ,  $\sin()$  в соответствии с вариантами задания.

В качестве образца рекомендуется взять стандартные библиотечные процедуры используемой языковой среды.

Второй способ заключается в использовании укороченных степенных рядов со специально подобранными (оптимизированными) значениями коэффициентов, обеспечивающими в заданном диапазоне применения примерно одинаковые (**0.0001**) значения абсолютной и относительной точностей:

$$e^x \approx \sum_{k=0}^7 e_k \cdot x^k, \quad |x| \leq 1$$

$$e_k = (0.99999, 1.0, 0.5, 0.16666, 0.04163, 0.00833, 0.00144, 0.0002), \quad k = 0, \dots, 7$$

$$\lg x \approx \sum_{k=1}^4 l_k \cdot \left( \frac{x-1}{x+1} \right)^{2k-1}, \quad 1 \leq x < \sqrt{10}$$

$$l_k = (0.86855, 0.29115, 0.15361, 0.21139), \quad k = 1, \dots, 4$$

$$\sin x \approx \sum_{k=1}^5 s_k \cdot x^{2k-1}, \quad |x| \leq \pi/2$$

$$s_k = (1.0, -0.166666, 0.00833, -0.00019, 0.000002), \\ k = 1, \dots, 5$$

Третий способ предполагает использование табличного разложения рассматриваемых функций в степенной ряд:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, |x| < \infty$$

$$\begin{aligned} \lg x &= \frac{2}{\ln 10} \cdot \sum_{k=1}^{\infty} \frac{(x-1)^{2k-1}}{(2k-1)(x+1)^{2k-1}} = \\ &= \frac{2}{\ln 10} \cdot \left[ \frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots \right], x > 0 \end{aligned}$$

$$\sin x = \sum_{k=1}^{\infty} (-1)^{k-1} \cdot \frac{x^{2k-1}}{(2k-1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, |x| < \infty$$

Для третьего способа расчета необходимо получить число просуммированных членов ряда, обеспечивающих вычисление искомой функции с задаваемой абсолютной или относительной степенями точности. Оформить выводимые на экран видеотерминала результаты в виде таблицы. Оценить точность полученных результатов. Объяснить наблюдаемые отклонения с точки зрения физического представления данных цифровой ЭВМ.

### Варианты задания

1.  $\exp(1 + 2x - 3x^2)$  2.  $\lg|2x + \operatorname{arctg} x|$  3.  $\sin(7\sin x - 0.2)$
4.  $\exp(0.1 + 3\cos x^2)$  5.  $\lg|2.51x + \lg x^2|$  6.  $\sin(1 - x + \operatorname{tg} x)$
7.  $\exp(0.4 + x + x^3)$  8.  $\lg|0.24 + 3\sin x|$  9.  $\sin(1 + x + 2e^x)$
10.  $\exp(3x - \operatorname{arctg} x)$  11.  $\lg|1 + 3x - \operatorname{tg} x|$  12.  $\sin(1 - 2x + x^3)$
13.  $\exp(2 + x - \ln|x|)$  14.  $\lg|1 + 8x - 2e^x|$  15.  $\sin(2.4x - \lg x^2)$
16.  $\exp(1 - x + 3\operatorname{tg} x)$  17.  $\lg|0.2 - 5\cos x|$  18.  $\sin(1 - x + 2x^2)$
19.  $\exp(0.12 + 6\sin x)$  20.  $\lg|2 + 2x + 2x^2|$  21.  $\sin(0.33 + \ln|x|)$
22.  $\exp(0.56x + \lg x^2)$  23.  $\lg|3.2x + 0.2x^3|$  24.  $\sin(x + \operatorname{arctg} x)$
25.  $\exp(1 + 5x + 2e^x)$  26.  $\lg|1 + \ln|x| + x|$  27.  $\sin(0.7x + 2x^4)$
28.  $\exp(2.2x + 0.3x^5)$  29.  $\lg|3 - 6x - 5x^2|$  30.  $\sin(7\cos x^3 - 1)$

**Пример 4.** Для аргумента  $\mathbf{a} = \mathbf{x}^2 + \sqrt{\mathbf{x}} - 2$  требуется составить программу для ЭВМ, позволяющую вычислить тремя способами таблицы функций  $\mathbf{exp}(\mathbf{a})$ ,  $\mathbf{lg}|\mathbf{a}|$ ,  $\mathbf{sin}(\mathbf{a})$  в заданном диапазоне изменения независимой переменной  $\mathbf{x}$  от начального значения  $\mathbf{x}_l$  с шагом изменения  $\mathbf{x}_s$  до конечного значения  $\mathbf{x}_u$ .

Использование стандартных библиотечных процедур  $\mathbf{exp}()$ ,  $\mathbf{lg}()$ ,  $\mathbf{sin}()$  очевидно. Укороченные степенные ряды со специально подобранными значениями коэффициентов легко получаются через использование счетной циклической алгоритмической структуры. Трудности могут возникнуть лишь при использовании табличных разложений функций в степенные ряды, поскольку вычисление очередных  $\mathbf{k} - \mathbf{x}$  членов  $\mathbf{z}_k$  при достаточно большом  $\mathbf{k}$  может вызвать переполнение разрядной сетки ЭВМ. Это преодолевается использованием рекуррентных формул для вычисления последующих членов ряда через предыдущие:

$$\mathbf{exp}() : \quad \mathbf{z}_{k+1} = \frac{\mathbf{x}^{k+1}}{(\mathbf{k} + 1)!} = \frac{\mathbf{x}^k}{\mathbf{k}!} \cdot \frac{\mathbf{x}}{\mathbf{k} + 1} = \mathbf{z}_k \cdot \frac{\mathbf{x}}{\mathbf{k} + 1} ;$$

$$\begin{aligned} \mathbf{lg}() : \quad \mathbf{z}_{k+1} &= \frac{2}{\ln 10} \cdot \frac{(\mathbf{x} - 1)^{2k+1}}{(2k + 1)(\mathbf{x} + 1)^{2k+1}} = \\ &= \frac{2}{\ln 10} \cdot \frac{(\mathbf{x} - 1)^{2k-1}}{(2k - 1)(\mathbf{x} + 1)^{2k-1}} \cdot \frac{2k - 1}{2k + 1} \cdot \left(\frac{\mathbf{x} - 1}{\mathbf{x} + 1}\right)^2 = \\ &= \mathbf{z}_k \cdot \frac{2k - 1}{2k + 1} \cdot \left(\frac{\mathbf{x} - 1}{\mathbf{x} + 1}\right)^2 ; \end{aligned}$$

$$\begin{aligned}
\sin() : \quad z_{k+1} &= (-1)^k \cdot \frac{x^{2k+1}}{(2k+1)!} = \\
&= -(-1)^{k-1} \cdot \frac{x^{2k-1}}{(2k-1)!} \cdot \frac{x^2}{2k(2k+1)} = \\
&= -z_k (-1)^{k-1} \cdot \frac{x^2}{2k(2k+1)}.
\end{aligned}$$

Невозможно суммирование бесконечно большого количества членов ряда. Поэтому вычисление с заданными абсолютной  $\epsilon$  или относительной  $\delta$  степенями точности подразумевает критерии останова процесса суммирования  $|z_k| < \epsilon$  и  $|z_k/s_k| < \delta$  соответственно для достижения требуемых значений абсолютной и относительной погрешностей вычисления суммы ряда  $s_k$  на  $k$ -м шаге. Рассмотренный способ вычисления через табличное разложение функции в степенной ряд рекомендуется реализовать через использование циклической алгоритмической структуры общего вида.

```

C      ЗАДАНИЕ 4. Пример 4. Прогр. на языке FORTRAN.
C      PROGRAM T4
C
C      PARAMETER( QMIN = 1.E-38 )
C      REAL      E(0:7), L(4), S(5)
C      DATA     E / 0.99999, 1., 0.5, 0.16666,
C      *         0.04163, 8.33E-3, 1.44E-3, 2.E-4 /,
C      *         L / 0.86855, 0.29115, 0.15361, 0.21139 /,
C      *         S / 1., -0.166666, 8.33E-3, -1.9E-4, 2.E-6 /
C
C      5 WRITE(*,*) 'INPUT XL, XU, XS, DLT:'
C      READ(*,*) XL, XU, XS, DLT

```

```

C
N = (XU-XL)/XS + 0.5 ! Кол-во XS в [XL,XU]
DO 88 I = 0, N ! Начало цикла по X
  X = XL + XS*FLOAT( I )
  WRITE(*,*)
  WRITE(*,*) ' X =', X
C
  A = X**2+SQRT(X)-2. ! Аргумент
  J = 0 ! вычисляемых функций
C
  7 J = J + 1 ! Перебор вычисляемых функций
  A2 = 0.0 ! Абсолютные и относительные
  R2 = 0.0 ! погрешности
  A3 = 0.0 ! для укороченных рядов
  R3 = 0.0 ! и табличных разложений
C
  GOTO( 11, 22, 33, 88 ) J
C
  11 Y1 = EXP( A ) ! Стандартная функция
  WRITE(*,*) 'EXP(', A, ') =', Y1
C
  Z = 1.0
  Y2 = E(0) ! Укороченный ряд
  DO K = 1, 7
    Z = Z * A
    Y2 = Y2 + E(K) * Z
  END DO
C
  K = 1
  Z = 1.0
  Y3 = Z ! Табличное разложение

```

```

15      Z = Z * A / FLOAT( K )
      IF( ABS( Z/Y3 ) .LT. DLT )      GOTO 44
      Y3 = Y3 + Z
      K = K + 1
      GOTO 15

C
22      Q = ABS( A )
      IF( Q .LT. QMIN ) GOTO 7
      Y1 = ALOG10( Q ) ! Стандартная функция
      WRITE(*,*)' Lg( ', Q, ' ) =', Y1

C
      V = ( Q - 1. ) / ( Q + 1. )
      W = V * V
      Z = V
      Y2 = 0.0          ! Укороченный ряд
      DO  K = 1, 4
      Y2 = Y2 + L(K) * Z
      Z = Z * W
      END DO

C
      K = 1
      Z = V * 2. / ALOG( 10. )
      Y3 = Z          ! Табличное разложение
      IF( ABS( Y1 ) .LT. QMIN ) GOTO 66
25      M = 2 * K
      Z = Z * W * FLOAT( M-1 ) / FLOAT( M+1 )
      IF( ABS( Z/Y3 ) .LT. DLT ) GOTO 44
      Y3 = Y3 + Z
      K = K + 1
      GOTO 25

C

```

```

33      Y1 = SIN( A )      ! Стандартная функция
      WRITE(*,*) 'SIN(', A, ') =', Y1
C
      V = A
      W = V * V
      Z = V
      Y2 = 0.0             ! Укороченный ряд
      DO  K = 1, 5
          Y2 = Y2 + S(K) * Z
          Z = Z * W
      END DO
C
      K = 1
      Z = V
      Y3 = Z               ! Табличное разложение
      IF( ABS( Y1 ) .LT. QMIN ) GOTO 66
35      M = 2 * K
          Z = - Z * W / FLOAT( M * ( M + 1 ) )
          IF( ABS( Z/Y3 ) .LT. DLT ) GOTO 44
          Y3 = Y3 + Z
          K = K + 1
          GOTO 35
C
44      A2 = Y2 - Y1
          R2 = ABS( A2 / Y1 )
          A3 = Y3 - Y1
          R3 = ABS( A3 / Y1 )
C
66      WRITE(*,*) 'Y2 =', Y2,
*          ' A2 =', A2, ' R2 =', R2

```



```

        WRITE(*,*) 'Y3 =', Y3,
*           ' A3 =', A3, ' R3 =', R3
        WRITE(*,*) ' K =', K
        GOTO 7      ! Конец перебора вычисл.функц.
C
88  CONTINUE      ! Конец цикла по X
    GOTO 5
    END

```

```

INPUT  XL, XU, XS, DLT:
0. 1. 0.5 6.E-8

```

```

X = 0.000000E+00
EXP(   -2.000000 ) =  1.353353E-01
Y2 = 1.327900E-01  A2 =-2.545E-03  R2 = 1.881E-02
Y3 = 1.353353E-01  A3 = 0.000E+00  R3 = 0.000E+00
K = 16

```

```

Lg(    2.000000 ) =  3.010300E-01
Y2 = 3.010288E-01  A2 =-1.192E-06  R2 = 3.960E-06
Y3 = 3.010300E-01  A3 =-2.980E-08  R3 = 9.900E-08
K = 7

```

```

SIN(   -2.000000 ) = -9.092974E-01
Y2 =-9.099360E-01  A2 =-6.385E-04  R2 = 7.022E-04
Y3 =-9.092975E-01  A3 =-5.960E-08  R3 = 6.555E-08
K = 7

```

```

X = 5.000000E-01
EXP(   -1.042893 ) =  3.524336E-01
Y2 = 3.524245E-01  A2 =-9.090E-06  R2 = 2.579E-05
Y3 = 3.524336E-01  A3 = 0.000E+00  R3 = 0.000E+00
K = 12

```

```

Lg(      1.042893 ) =  1.823982E-02
Y2 = 1.823902E-02  A2 =-8.028E-07   R2 = 4.401E-05
Y3 = 1.823982E-02  A3 = 0.000E+00   R3 = 0.000E+00
K = 2
SIN(     -1.042893 ) = -8.638652E-01
Y2 =-8.638721E-01  A2 =-6.914E-06   R2 = 8.004E-06
Y3 =-8.638653E-01  A3 =-5.960E-08   R3 = 6.900E-08
K = 5

X =      1.000000
EXP( 0.000000E+00 ) =      1.000000
Y2 = 9.999900E-01  A2 =-1.001E-05   R2 = 1.001E-05
Y3 =      1.000000  A3 = 0.000E+00   R3 = 0.000E+00
K = 1
SIN( 0.000000E+00 ) =  0.000000E+00
Y2 = 0.000000E+00  A2 = 0.000E+00   R2 = 0.000E+00
Y3 = 0.000000E+00  A3 = 0.000E+00   R3 = 0.000E+00
K = 1

```

```

/* ЗАДАНИЕ 4. Пример 4. Программа на языке C. */

```

```

#include <stdio.h>
#include <math.h>

int main( void ) {

int i, j, k, m, n;

double a, a2, a3, dlt, q, qmin = 1.e-38, r2, r3,
       v, w, x, xl, xu, xs, y1, y2, y3, z;

```

```

double e[8] = { 0.99999, 1., 0.5, 0.16666,
               0.04163, 8.33e-3, 1.44e-3, 2.e-4 };

double l[4] = { 0.86855, 0.29115,
               0.15361, 0.21139 };

double s[5] = { 1.0, -0.166666,
               8.33e-3, -1.9e-4, 2.e-6 };

l1: printf( "input xl, xu, xs, dlt :\n" );
scanf( "\f %lg %lg %lg %lg",
       &xl, &xu, &xs, &dlt );

n = (xu-xl)/xs + 0.5; /* Кол-во xs в [xl,xu] */

for( i = 0; i <= n; i++ ) { /* Нач.цикла по x */

    x = xl + xs * (double)i;
    printf( "\n x =% e\n", x );

    a = x * x + sqrt( x ) - 2.; /* Аргумент */
    j = 0; /* вычисляемых функций */

    while( j < 3 ) { /* Перебор вычисл.функций */

        j++;
        a2 = 0.0; /* Абсолютные и относительные */
        r2 = 0.0; /* погрешности */
        a3 = 0.0; /* для укороченных рядов */
        r3 = 0.0; /* и табличных разложений */
    }
}

```

```

switch( j ) {

    case 1:
        y1 = exp( a ); /* Стандартная функция */
        printf( "exp(% e ) =% e \n", a, y1 );

        z = 1.0;
        y2 = e[0];          /* Укороченный ряд */
        for( k = 1; k <= 7; k++ ) {
            z *= a;
            y2 += e[k] * z;
        }

        k = 1;
        z = 1.0;
        y3 = z;          /* Табличное разложение */
        while( 1 ) {
            z *= a / (double)k;
            if( fabs( z/y3 ) < dlt ) break;
            y3 += z;
            k++;
        } break;

    case 2:
        q = fabs( a );
        if( q < qmin ) continue;
        y1 = log10( q ); /* Стандартная функция */
        printf( " lg(% e ) =% e \n", a, y1 );

```

```

v = ( q - 1.) / ( q + 1.);
w = v * v;
z = v;
y2 = 0.0;          /* Укороченный ряд */
for( k = 0; k <= 3; k++ ) {
    y2 += l[k] * z;
    z *= w;
}

k = 1;
z = v * 2. / log( 10. );
y3 = z;          /* Табличное разложение */
if( fabs( y1 ) < qmin ) goto l2;
while( 1 ) {
    m = 2 * k;
    z *= w*(double)( m-1 )/(double)( m+1 );
    if( fabs( z/y3 ) < dlt ) break;
    y3 += z;
    k++;
} break;

case 3:
y1 = sin( a ); /* Стандартная функция */
printf( "sin(% e ) =% e \n", a, y1 );

v = a;
w = v * v;
z = v;
y2 = 0.0;          /* Укороченный ряд */
for( k = 0; k <= 4; k++ ) {
    y2 += s[k] * z;

```

```

        z *= w;
    }

    k = 1;
    z = v;
    y3 = z;          /* Табличное разложение */
    if( fabs( y1 ) < qmin ) goto l2;
    while( 1 ) {
        m = 2 * k;
        z *= -w / (double)( m * ( m+1 ) );
        if( fabs( z/y3 ) < dlt ) break;
        y3 += z;
        k++;
    } break;

    default: continue;
}

a2 = y2 - y1;
r2 = fabs( a2 / y1 );
a3 = y3 - y1;
r3 = fabs( a3 / y1 );

l2: printf( "y2 =% e  a2 =% .4e  r2 =% .4e\n",
           y2,      a2,      r2 );
      printf( "y3 =% e  a3 =% .4e  r3 =% .4e\n",
           y3,      a3,      r3 );
      printf( " k =% d\n", k );

    }          /* Конец перебора вычисляемых функций */
}            /* Конец цикла по независ. перемен. x */

```

```

        goto l1;
    }

input  xl, xu, xs, dlt :
0. 1. 0.5 1.12e-16

    x = 0.000000e+00
exp(-2.000000e+00 ) = 1.353353e-01
y2 = 1.327900e-01  a2 =-2.5453e-03  r2 = 1.8807e-02
y3 = 1.353353e-01  a3 = 5.5511e-17  r3 = 4.1018e-16
    k = 25
    lg(-2.000000e+00 ) = 3.010300e-01
y2 = 3.010288e-01  a2 =-1.1982e-06  r2 = 3.9804e-06
y3 = 3.010300e-01  a3 = 0.0000e+00  r3 = 0.0000e+00
    k = 16
    sin(-2.000000e+00 ) =-9.092974e-01
y2 =-9.099360e-01  a2 =-6.3857e-04  r2 = 7.0227e-04
y3 =-9.092974e-01  a3 = 0.0000e+00  r3 = 0.0000e+00
    k = 12

    x = 5.000000e-01
exp(-1.042893e+00 ) = 3.524335e-01
y2 = 3.524244e-01  a2 =-9.1377e-06  r2 = 2.5928e-05
y3 = 3.524335e-01  a3 = 1.6653e-16  r3 = 4.7252e-16
    k = 19
    lg(-1.042893e+00 ) = 1.823984e-02
y2 = 1.823904e-02  a2 =-8.0318e-07  r2 = 4.4034e-05
y3 = 1.823984e-02  a3 =-3.4694e-18  r3 = 1.9021e-16
    k = 5
    sin(-1.042893e+00 ) =-8.638652e-01
y2 =-8.638721e-01  a2 =-6.8685e-06  r2 = 7.9509e-06

```

y3 = -8.638652e-01   a3 = 0.0000e+00   r3 = 0.0000e+00  
k = 9

x = 1.000000e+00  
exp( 0.000000e+00 ) = 1.000000e+00  
y2 = 9.999900e-01   a2 = -1.0000e-05   r2 = 1.0000e-05  
y3 = 1.000000e+00   a3 = 0.0000e+00   r3 = 0.0000e+00  
k = 1  
sin( 0.000000e+00 ) = 0.000000e+00  
y2 = 0.000000e+00   a2 = 0.0000e+00   r2 = 0.0000e+00  
y3 = 0.000000e+00   a3 = 0.0000e+00   r3 = 0.0000e+00  
k = 1



## ЗАДАНИЕ 5

На основе счетной циклической алгоритмической структуры с требуемым уровнем вложенности составить программу для ЭВМ, позволяющую найти значения величин, определяемых пунктами **a)**, **b)**, **c)** вариантов задания для матрицы, включающей **m** строк и **n** столбцов элементов, числовые значения которых заданы произвольным образом. Вывести на экран видеотерминала исходную матрицу и полученные результаты.

### Варианты задания

**1. m = 5; n = 5.** Найти: **a)** сумму больших значения **3** элементов выше главной диагонали; **b)** произведение отрицательных элементов; **c)** минимальный элемент побочной диагонали.

**2. m = 6; n = 5.** Найти: **a)** максимальный отрицательный элемент 4-го столбца; **b)** сумму положительных элементов ниже 3-й строки; **c)** индексы минимального элемента.

**3. m = 5; n = 5.** Найти: **a)** произведение ненулевых элементов; **b)** минимальный больший значения **3** элемент на главной диагонали; **c)** индексы и значение минимального положительного элемента.

**4. m = 6; n = 4.** Найти: **a)** сумму неповторяющихся положительных элементов 2-го столбца и 3-й строки; **b)** минимальный положительный элемент; **c)** максимальный отрицательный элемент.

**5. m = 6; n = 5.** Найти: **a)** максимальные элементы каждого столбца и образовать из них вектор; **b)** индексы ну-

левых элементов; **с**) минимальный положительный элемент 5-й строки.

**6.  $m = 5$ ;  $n = 5$ .** Найти: **а**) сумму больших значения **1** элементов; **б**) количество отрицательных элементов главной диагонали; **с**) максимальный элемент побочной диагонали.

**7.  $m = 5$ ;  $n = 5$ .** Найти: **а**) сумму положительных элементов главной диагонали; **б**) минимальный элемент 3-й строки; **с**) индексы и количество отрицательных элементов.

**8.  $m = 6$ ;  $n = 4$ .** Найти: **а**) произведение элементов, модуль которых больше значения **2**; **б**) количество отрицательных элементов; **с**) индексы максимального элемента.

**9.  $m = 6$ ;  $n = 5$ .** Найти: **а**) минимальный положительный элемент левее 4-го столбца; **б**) произведение отрицательных элементов ниже 3-й строки; **с**) индексы нулевых элементов.

**10.  $m = 6$ ;  $n = 4$ .** Найти: **а**) максимальный отрицательный элемент выше 4-й строки; **б**) сумму больших значения **5** элементов четных столбцов; **с**) индексы максимального элемента.

**11.  $m = 5$ ;  $n = 5$ .** Найти: **а**) сумму и количество неотрицательных элементов 3-го столбца; **б**) минимальный положительный элемент ниже главной диагонали; **с**) произведение ненулевых элементов.

**12.  $m = 5$ ;  $n = 4$ .** Найти: **а**) сумму положительных элементов каждой строки и образовать из них вектор; **б**) минимальный элемент ниже 2-й строки; **с**) индексы любого нулевого элемента.

**13.  $m = 5$ ;  $n = 5$ .** Найти: **а**) минимальный положительный элемент выше главной диагонали; **б**) количество отрицательных элементов; **с**) индексы нулевых элементов побочной диагонали.

**14.  $m = 5$ ;  $n = 5$ .** Найти: **a)** максимальный элемент главной диагонали; **b)** произведение и количество положительных элементов выше главной диагонали; **c)** сумму отрицательных элементов.

**15.  $m = 5$ ;  $n = 5$ .** Найти: **a)** сумму и количество отрицательных элементов побочной диагонали; **b)** произведение больших значения 4 элементов; **c)** число отрицательных элементов ниже 3-й строки.

**16.  $m = 5$ ;  $n = 5$ .** Найти: **a)** произведение и количество отрицательных элементов выше побочной диагонали; **b)** индексы меньших значения -5 элементов; **c)** максимальный отрицательный элемент.

**17.  $m = 6$ ;  $n = 4$ .** Найти: **a)** сумму и количество ненулевых элементов; **b)** произведение неповторяющихся положительных элементов правее 2-го столбца или ниже 3-й строки; **c)** индексы любого нулевого элемента.

**18.  $m = 5$ ;  $n = 5$ .** Найти: **a)** минимальный положительный элемент; **b)** сумму и количество неотрицательных элементов правее 2-го столбца; **c)** максимальный меньший значения 7 элемент главной диагонали.

**19.  $m = 6$ ;  $n = 5$ .** Найти: **a)** максимальный элемент на пересечении четных строк и нечетных столбцов; **b)** произведение ненулевых элементов с суммой индексов не более 5; **c)** количество нулевых элементов.

**20.  $m = 6$ ;  $n = 4$ .** Найти: **a)** сумму и количество отрицательных элементов четных строк; **b)** произведение ненулевых элементов 3-го столбца; **c)** минимальный больший значения -6 элемент.

**21.  $m = 6$ ;  $n = 5$ .** Найти: **a)** максимальный отрицательный элемент левее 4-го столбца; **b)** количество больших значения 3 элементов; **c)** индексы минимального элемента.

**22.  $m = 5$ ;  $n = 5$ .** Найти: **a)** минимальный положительный элемент ниже главной диагонали; **b)** сумму меньших значения **-2** элементов; **c)** индексы минимального элемента побочной диагонали.

**23.  $m = 5$ ;  $n = 5$ .** Найти: **a)** сумму и количество элементов, модуль которых больше значения **2** и меньше значения **6**; **b)** индексы нулевых элементов главной и побочной диагоналей; **c)** индексы максимального элемента.

**24.  $m = 6$ ;  $n = 5$ .** Найти: **a)** сумму элементов с произведением индексов не более **12**; **b)** максимальный элемент нечетных столбцов; **c)** количество отрицательных элементов.

**25.  $m = 6$ ;  $n = 4$ .** Найти: **a)** количество перемен знаков элементов четных строк; **b)** сумму отрицательных элементов на пересечении нечетных строк и четных столбцов; **c)** минимальный элемент.

**26.  $m = 5$ ;  $n = 5$ .** Найти: **a)** количество меньших значения **4** элементов выше главной, но ниже побочной диагонали; **b)** сумму индексов нулевых элементов; **c)** максимальный элемент 5-го столбца.

**27.  $m = 6$ ;  $n = 4$ .** Найти: **a)** количество перемен знаков элементов на пересечении четных строк и столбцов; **b)** сумму элементов нечетных строк; **c)** индексы минимального элемента.

**28.  $m = 5$ ;  $n = 5$ .** Найти: **a)** максимальный элемент выше главной и ниже побочной диагонали; **b)** сумму элементов на пересечении нечетных строк и столбцов; **c)** количество положительных элементов.

**29.  $m = 6$ ;  $n = 5$ .** Найти: **a)** произведение индексов максимальных элементов строк; **b)** сумму меньших значения **-1** элементов ниже 3-й строки; **c)** количество нулевых

элементов.

**30.**  $m = 6$ ;  $n = 4$ . Найти: **a)** сумму отрицательных больших значения **-8** элементов; **b)** произведение индексов нулевых элементов 2-й и 3-й строк; **c)** количество положительных элементов.

**Пример 5а.** Дана матрица, включающая  $m = 5$  строк и  $n = 5$  столбцов элементов, числовые значения которых заданы произвольным образом. Требуется найти:

**a)** количество перемен знаков и нулевые элементы на главной диагонали;

**b)** произведение элементов побочной диагонали, модуль значения которых больше **3**;

**c)** сумму и количество отрицательных элементов в 3-м столбце;

**d)** сумму отрицательных элементов каждой строки и образовать из них вектор;

**e)** максимальный элемент выше 4-й строки и его индексы;

**f)** минимальный положительный элемент на пересечении четных строк и нечетных столбцов и его индексы.

Искомый результат может быть достигнут последовательным перебором элементов матрицы по строкам и столбцам на основе счетной циклической алгоритмической структуры. Последовательность перебора может быть задана по желанию пользователя. Обработка элементов матрицы в соответствии с требованиями задания должна составить тело вложенной циклической структуры.

**C**      **ЗАДАНИЕ 5. Пример 5а. Progr. на яз. FORTRAN.**  
          **PROGRAM T5A**

**C**

```

CHARACTER C
R E A L   A(5,5), Sd(5)
C
D A T A   A / -1., 3.,-7., 5., 4.,
*          1., 6.,-2., 0., 8.,
*          -5., 1.,-9.,-2., 1.,
*          6., 4.,-2., 0., 7.,
*          2., 9.,-3.,-1., 8. /
10  WRITE(*,11)
11  FORMAT(2X,'Новые значения элементов A(5,5)?',
*         ' [Y/N D:N] : ', $)
    READ(*,15) C
15  FORMAT(A1)
    IF( C .NE. 'Y' .AND. C .NE. 'y' )      GOTO 20
C
    DO I = 1 , 5
      READ(*,*) ( A(I,J), J = 1 , 5 )
    END DO
C
20  WRITE(*,*)
    DO I = 1 , 5
      WRITE(*,22) ( A(I,J), J = 1 , 5 )
22  FORMAT(5G10.3)
    END DO
C
    Ea = A(1,1)  ! Инициализация значения знака
    Ka = 0       ! и количества его перемен а).
C
    Pb = 1.0     ! Инициализ.знач.произвед.  b).
C
    Sc = 0.0     ! Инициализация значения суммы

```

```

      Kc = 0          ! и количества c).
C
      Ee = -1.E38    ! Инициализация знач. макс. эл-та
      Ie = 0         ! и его
      Je = 0         ! индексов e).
C
      Ef = 1.E38    ! Инициализация знач. мин. эл-та
      If = 0         ! и его
      Jf = 0         ! индексов f).
C
      WRITE(*,*)
      DO I = 1 , 5          ! Цикл по строкам
C
          IF( ABS(A(I,I)) .LT. 1.E-38 ) ! Главная
*          WRITE(*,25) I, I, A(I,I) ! диагональ
25          FORMAT(' A('I1,', 'I1,') = ',G10.3,/)
C
          IF( Ea * A(I,I) .LT. 0.0 ) THEN
              Ea = A(I,I)          ! Перемена
              Ka = Ka + 1          ! знака
          END IF
C
          J = 6 - I                ! Побочная диагональ
          IF( ABS(A(I,J)) .GT. 3.0 ) Pb = Pb * A(I,J)
C
          IF( A(I,3) .LT. 0.0 ) THEN ! 3-й столбец
              Sc = Sc + A(I,3)
              Kc = Kc + 1
          END IF
C
      Sd(I) = 0.0 ! Инициализ. компонент вект. d).

```

```

C      DO J = 1 , 5          ! Цикл по столбцам
C
      IF( A(I,J) .LT. 0.0 )
*        Sd(I) = Sd(I) + A(I,J)
C
      IF( I .LE. 3 ) THEN ! Выше 4-й строки
        IF( A(I,J) .GT. Ee ) THEN
          Ee = A(I,J)
          Ie = I
          Je = J
        END IF
      END IF
C
      IF( (I/2)*2 .EQ. I ! Проверка чет./нечет.
*        .AND. (J/2)*2 .NE. J ) THEN
        IF( A(I,J) .GT. 0.0
*        .AND. A(I,J) .LT. Ef ) THEN
          Ef = A(I,J)
          If = I
          Jf = J
        END IF
      END IF
C
      END DO          ! Конец цикла по столбцам
      END DO          ! Конец цикла по строкам
C
      WRITE(*,35) Ka, Pb
35     FORMAT(' Ka =',I2,4X,'Pb =',G10.3,/)
C
      WRITE(*,55) Kc, Sc

```



```

55  FORMAT(' Kc =',I2,4X,'Sc =',G10.3,/)
C
    WRITE(*,75) Sd
75  FORMAT(' Sd :',5G9.2,/)
C
    WRITE(*,95) Ie, Je, Ee
    WRITE(*,95) If, Jf, Ef
95  FORMAT(' A('I1,', 'I1,') = ',G10.3,/)
C
    GOTO 10
    END

```

Новые значения элементов A(5,5)? [Y/N D:N] :

-1.00	1.00	-5.00	6.00	2.00
3.00	6.00	1.00	4.00	9.00
-7.00	-2.00	-9.00	-2.00	-3.00
5.00	.000	-2.00	.000	-1.00
4.00	8.00	1.00	7.00	8.00

A(4,4) = .000

Ka = 3      Pb = -144.

Kc = 3      Sc = -16.0

Sd : -6.0      .00      -23.      -3.0      .00

A(2,5) = 9.00

A(2,3) = 1.00

```
/* ЗАДАНИЕ 5. Пример 5а. Программа на языке С. */
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main( void ) {
```

```
char c;
```

```
int i, j, Ie, Je, If, Jf, Ka, Kc;
```

```
double Ea, Ee, Ef, Pb, Sc, Sd[5];
```

```
double a[5][5] = { -1., 1., -5., 6., 2.,  
                  3., 6., 1., 4., 9.,  
                  -7., -2., -9., -2., -3.,  
                  5., 0., -2., 0., -1.,  
                  4., 8., 1., 7., 8. };
```

```
ll: printf( "\nНовые значения элементов a[5][5]" );
```

```
printf( " ? [y/n d:n] : " );
```

```
scanf ( "\f %c", &c );
```

```
if( c=='y' || c=='Y' ) {
```

```
    for( i = 0; i < 5; i++ ) {
```

```
        for( j = 0; j < 5; j++ ) {
```

```
            scanf( "%lg", &a[i][j] );
```

```
        }
```

```
    }
```

```
}
```

```

printf( "\n" );
for( i = 0; i < 5; i++ ) {
    for( j = 0; j < 5; j++ ) {
        printf( "%9.1f", a[i][j] );
    } printf( "\n" );
}

Ea = a[0][0]; /* Инициализац. значения знака */
Ka = 0;      /* и количества его перемен а).*/

Pb = 1.0;    /* Инициализ.знач.произвед. b).*/

Sc = 0.0;    /* Инициализация значения суммы*/
Kc = 0;      /* и количества с). */

Ee = -1.E308; /* Инициализац.знач.макс.эл-та */
Ie = 0;      /* и его */
Je = 0;      /* индексов е). */

Ef = 1.E308; /* Инициализац.знач. мин.эл-та */
If = 0;      /* и его */
Jf = 0;      /* индексов f). */

for( i = 0; i < 5; i++ ) { /* Цикл по строкам*/

    if( fabs( a[i][i] ) < 1.e-38 ) { /* Главная */
        j = i + 1;                /* диагональ */
        printf( "\nA(%1d,%1d)", j, j );
        printf( " =%5.1f\n", a[i][i] );
    }
}

```

```

if( Ea * a[i][i] < 0.0 ) {
    Ea = a[i][i];          /* Перемена */
    Ka++;                  /* знака */
}
j = 4 - i;                /* Побочная диагональ */
if( fabs( a[i][j] ) > 3.0 ) Pb *= a[i][j];

if( a[i][2] < 0.0 ) {     /* 3-й столбец */
    Sc += a[i][2];
    Kc++;
}

Sd[i] = 0.0; /* Инициализ.компон.вект. d). */

for( j = 0; j < 5; j++ ){ /*Цикл по столбцам*/

    if( a[i][j] < 0.0 ) Sd[i] += a[i][j];

    if( i < 3 ) {         /* Выше 4-й строки */
        if( a[i][j] > Ee ) {
            Ee = a[i][j];
            Ie = i;
            Je = j;
        }
    }
}

if( (i/2)*2 != i /* Проверка чет./нечет. */
    && (j/2)*2 == j ) {
    if( a[i][j] > 0.0
        && a[i][j] < Ef ) {
        Ef = a[i][j];
    }
}

```

```

        If = i;
        Jf = j;
    }
}
} /* Конец цикла по столбцам */
} /* Конец цикла по строкам */

printf( "\nKa =%2d   Pb =%5.1f\n", Ka, Pb );
printf( "\nKc =%2d   Sc =%5.1f\n", Kc, Sc );
printf( "\nSd : " );
for( i = 0; i < 5; i++ ) {
    printf( "%8.1f", Sd[i] );
} printf( "\n" );
printf( "\nA(%1d,%1d) =%5.1f\n",Ie+1,Je+1,Ee );
printf( "\nA(%1d,%1d) =%5.1f\n",If+1,Jf+1,Ef );

goto ll;
}

```

Новые значения элементов a[5][5] ? [y/n d:n] : n

-1.0	1.0	-5.0	6.0	2.0
3.0	6.0	1.0	4.0	9.0
-7.0	-2.0	-9.0	-2.0	-3.0
5.0	0.0	-2.0	0.0	-1.0
4.0	8.0	1.0	7.0	8.0

A(4,4) = 0.0

Ka = 3 Pb =-144.0

Kc = 3 Sc =-16.0

Sd :        -6.0        0.0        -23.0        -3.0        0.0

A(2,5) = 9.0

A(2,3) = 1.0

**Пример 5b.** Умножение сцепленных прямоугольных матриц. Матрицы  $\mathbf{A}_{ML} = [a_{ik}]$  и  $\mathbf{B}_{LN} = [b_{kj}]$  называются сцепленными, если число столбцов первой из них равно числу строк второй ( параметр  $\mathbf{L}$  ). Их произведением будет прямоугольная матрица  $\mathbf{C}_{MN} = [c_{ij}]$ , каждый элемент которой получается в виде скалярного произведения  $i$ -й вектор-строки матрицы  $\mathbf{A}_{ML}$  на  $j$ -й вектор-столбец матрицы  $\mathbf{B}_{LN}$  :

$$c_{ij} = \sum_{k=1}^L a_{ik} b_{kj} .$$

Написать программу для ЭВМ, позволяющую найти произведение двух сцепленных матриц размерностей  $\mathbf{M} \times \mathbf{L}$  и  $\mathbf{L} \times \mathbf{N}$ , числовые значения элементов которых заданы произвольным образом.

```
C        ЗАДАНИЕ 5. Пример 5b. Прогр. на яз. FORTRAN.
C        PROGRAM T5B
C
C        PARAMETER( M = 2, L = 4, N = 3 )
C
C        DIMENSION A(M,L), B(L,N), C(M,N)
C
C        WRITE(*,*)
```

```

WRITE(*,*) 'A(M,L):'
DO I = 1 , M
  READ(*,*) ( A(I,J), J = 1 , L )
  WRITE(*,55) ( A(I,J), J = 1 , L )
END DO
C
WRITE(*,*)
WRITE(*,*) 'B(L,N):'
DO I = 1 , L
  READ(*,*) ( B(I,J), J = 1 , N )
  WRITE(*,55) ( B(I,J), J = 1 , N )
END DO
C
WRITE(*,*)
WRITE(*,*) 'C(M,N):'
DO I = 1, M
  DO J = 1 , N
    S = 0.0 ! C(I,J) = 0.0
    DO K = 1, L
      S = S + A(I,K) * B(K,J)
      ! C(I,J) = C(I,J) + A(I,K) * B(K,J)
    END DO
    C(I,J) = S
  END DO
  WRITE(*,55) ( C(I,J), J = 1 , N )
55  FORMAT(5G10.3)
END DO
C
STOP
END

```

```
A(M,L):  
-4.00   -3.00   2.00   1.00  
 3.00   -2.00   .000   3.00
```

```
B(L,N):  
 .000   .000   -4.00  
 3.00  -2.00  -3.00  
 3.00  -3.00   4.00  
-2.00   .000   1.00
```

```
C(M,N):  
-5.00   .000   34.0  
-12.0   4.00   -3.00
```

```
/* ЗАДАНИЕ 5. Пример 5b. Программа на языке C.*/  
/* Динамич.захват памяти под 2-мерные массивы */
```

```
#include <alloc.h>  
#include <stdio.h>  
  
int main(void) {  
  
    int M = 2, L = 4, N = 3, i, j, k, Kerr;  
  
    double **A, **B, **C, S;  
  
    Kerr = 1;  
    A = (double **)calloc( M, sizeof(double) );  
    if( !A ) return Kerr;
```



```

for( i = 0; i < M ; i++ ) {
    A[i] = (double *)calloc( L, sizeof(double) );
    if( !A[i] ) {
        while( i >= 0 ) free(A[i--]);
        free(A);
        return Kerr;
    }
}

Kerr = 2;
B = (double **)calloc( L, sizeof(double) );
if( !B ) goto Free_A;
for( i = 0; i < L ; i++ ) {
    B[i] = (double *)calloc( N, sizeof(double) );
    if( !B[i] ) {
        while( i >= 0 ) free(B[i--]);
        free(B);
        goto Free_A;
    }
}

Kerr = 3;
C = (double **)calloc( M, sizeof(double) );
if( !C ) goto Free_B;
for( i = 0; i < M ; i++ ) {
    C[i] = (double *)calloc( N, sizeof(double) );
    if( !C[i] ) {
        while( i >= 0 ) free(C[i--]);
        free(C); goto Free_B;
    }
}

```

```

printf( "\nMatrix A[M][L]:\n" );
for( i = 0; i < M ; i++ ) {
    for( j = 0; j < L ; j++ ) {
        scanf( "%lg", &S );
        A[i][j] = S;
        printf( "%9.2f", A[i][j] );
    } printf( "\n" );
}

printf( "\nMatrix B[L][N]:\n" );
for( i = 0; i < L ; i++ ) {
    for( j = 0; j < N ; j++ ) {
        scanf( "%lg", B[i] + j );
        printf( "%9.2f", B[i][j] );
    } printf( "\n" );
}

printf( "\nMatrix C[M][N]:\n" );
for( i = 0; i < M ; i++ ) {
    for( j = 0; j < N ; j++ ) {
        S = 0.0; /* C[i][j] = 0.0; */
        for( k = 0; k < L ; k++ ) {
            S += A[i][k] * B[k][j];
            /* C[i][j] += A[i][k] * B[k][j]; */
        }
        C[i][j] = S;
        printf( "%9.2f", C[i][j] );
    } printf( "\n" );
}

```

```

Kerr = 0;
for( i = M; i >= 0; i-- ) free(C[i]); free(C);

Free_B:
for( i = L; i >= 0; i-- ) free(B[i]); free(B);

Free_A:
for( i = M; i >= 0; i-- ) free(A[i]); free(A);

printf( "\nKerr =%2d", Kerr );

return Kerr;
}

```

Matrix A[M] [L]:

-4.00	-3.00	2.00	1.00
3.00	-2.00	0.00	3.00

Matrix B[L] [N]:

0.00	0.00	-4.00
3.00	-2.00	-3.00
3.00	-3.00	4.00
-2.00	0.00	1.00

Matrix C[M] [N]:

-5.00	0.00	34.00
-12.00	4.00	-3.00

Kerr = 0

## ЗАДАНИЕ 6

На основе счетной циклической алгоритмической структуры с требуемым уровнем вложенности составить программу для ЭВМ, позволяющую получить на экране видеотерминала квадратную символьную матрицу. Варианты задания показывают вид такой матрицы с числом строк и столбцов  $N = 11$ . Продемонстрировать работу программы при значениях  $N = 1 - 20$ .

### Варианты задания

<pre> * * * * * * * A A A A * *  * A A A A * *   * A A A A * *    * A A A A * 1. *     * A A A A * *      *  * A A A * *       *   * A A * *        *    * A * * *           * * * * * * * </pre>	<pre> * * * * * *      * *      * *      * *      * 2. * * * * * * * C C C * D D D D * *  * C C * D D D D * *   * C * D D D D * *    * * D D D D * * * * * * </pre>
---	---

```

* * * * *
*           * E E *
*           * E E * F *
*           * E E * F F *
*           * E E * F F *
3. * * E E * F F * *
* * E E * F F *
* E E * F F *
* E * F F *
* * F F *
* * * * *

```

```

* * * * *
* *           *
* *           * G *
* *           * G G *
* *           * G G G *
4. * * * * * G G G G *
* G G G G G G G G *
* G G G G G G G G *
* G G G G G G G G *
* G G G G G G G G *
* * * * *

```

```

* * * * *
*           * *
*           * H *
*           * H H *
* * H H H *
5. * * * * *
*           * O O O O *
*           * O O O O *
*           * O O O O *
*           * O O O O *
* * * * *

```

```

* * * * *
* * P P P P P P P *
* * P P P P P *
* * P P P *
* * P *
6. * * * * *
*           * Q Q Q Q *
*           * Q Q Q Q *
*           * Q Q Q Q *
*           * Q Q Q Q *
* * * * *

```

```

* * * * *
* *      * S S S S *
* R *      * S S S S *
* R R *      * S S S S *
* R R R * * S S S S *
7. * R R R R * * * * *
* R R R R R *      *
* R R R R R R *      *
* R R R R R R R * *
* R R R R R R R R * *
* * * * *
* * * * *
* * T T T T T T T * *
* * T T T T T * U *
* * T T T * U U *
* * T * U U U *
8. * * U U U U *
* * * U U U *
* * * U U *
* * * U *
* * * * *

```

```

* * * * *
* *      *      *
* V *      * W *
* V V *      * W W *
* V V V * * W W W *
9. * V V V V * W W W W *
* V V V V * * W W W *
* V V V V * * W W *
* V V V V *      * W *
* V V V V *      * *
* * * * *
* * * * *
* * * * * X X X * *
* * * * * X X * *
* * * * * X * *
* * * * * * *
10. * * * * * *
* * * * * *
* * * * * Y *
* * * * * Y Y *
* * * * * Y Y Y *
* * * * *

```

	* * * * *		* * * * *
	* * Z Z Z *		* I I I I I I I I *
	* * Z Z Z *		* I I I I I I I I *
	* * * * Z Z Z * * * *		* * * * I I I * * * *
11.	* Z Z Z Z Z Z Z Z Z *	12.	* * I I I * *
	* Z Z Z Z Z Z Z Z Z *		* * I I I * *
	* * * * Z Z Z * * * *		* * * * I I I * * * *
	* * Z Z Z *		* I I I I I I I I *
	* * Z Z Z *		* I I I I I I I I *
	* * * * *		* * * * *

	* * * * *		* * * * *
	* L L * * L L *		* T T T T T T T T *
	* L L * * L L *		* T T T T T T T T *
	* L L * * * * *		* * * * T T T * * * *
13.	* L L * * * * *	14.	* * T T T * *
	* L L * * * * *		* * T T T * *
	* L L * * * * *		* * T T T * *
	* L L L L L L L L L *		* * T T T * *
	* L L L L L L L L L *		* * T T T * *
	* * * * *		* * * * *

15.	<pre> * * * * * * X X *   * X X * * X X X * * X X X * * * X X X * X X X * * *   * X X X X X *   * *     * X X X *     * *   * X X X X X *   * * * X X X * X X X * * * X X X *   * X X X * * X X *     * X X * * * * * * </pre>	16.	<pre> * * * * * * Y Y *   * Y Y * * Y Y Y * * Y Y Y * * * Y Y Y * Y Y Y * * *   * Y Y Y Y Y *   * *     * Y Y Y *     * *   * Y Y Y *   * *     * Y Y Y *     * *   * Y Y Y *   * *     * Y Y Y *     * * * * * * </pre>
-----	--	-----	--

17.	<pre> * * * * * * Z Z Z Z Z Z Z Z Z * * Z Z Z Z Z Z Z Z Z * * * * * * Z Z Z * * *   * Z Z Z *   * *     * Z Z Z *     * *   * Z Z Z *   * * * Z Z Z * * * * * * Z Z Z Z Z Z Z Z Z * * Z Z Z Z Z Z Z Z Z * * * * * * </pre>	18.	<pre> * * * * * * 0 0 0 0 0 0 0 0 * * 0 * * * * * 0 * * 0 *           * 0 * * 0 *   * * *   * 0 * * 0 *   * 0 *   * 0 * * 0 *   * * *   * 0 * * 0 *           * 0 * * 0 * * * * * 0 * * 0 0 0 0 0 0 0 0 * * * * * * </pre>
-----	--	-----	--



```

* * * * *
* E E E E E E E E E *
* E * * * * * * * * *
* E *
* E * * * * * * * * *
19. * E E E E E E E E E *
* E * * * * * * * * *
* E *
* E * * * * * * * * *
* E E E E E E E E E *
* * * * *

```

```

* * * * *
* C C C C C C C C C *
* C C C C C C C C C *
* C C * * * * * * * *
* C C *
20. * C C *
* C C *
* C C * * * * * * * *
* C C C C C C C C C *
* C C C C C C C C C *
* * * * *

```

```

* * * * *
* S S S S S S S S S *
* S * * * * * * * * *
* S *
* S * * * * * * * * *
21. * S S S S S S S S S *
* * * * * * * * * S *
*
* * * * * * * * * S *
* S S S S S S S S S *
* * * * *

```

```

* * * * *
* M M * * * * M M *
* M M M * * M M M *
* M M M M * M M M M *
* M M M M M M M M M *
22. * M M * M M M * M M *
* M M * * M * * M M *
* M M * * * M M *
* M M * * * M M *
* M M * * * M M *
* * * * *

```

<pre> * * * * * * A A *   * B B * * A A *   * B B * * * * *   * * * * * 23. * * * * * *   * * * * * C C *   * D D * * C C *   * D D * * * * * * * * * * * </pre>	<pre> * * * * * * * * * * * F F F * * F F * * F * * * * * 24. * * * * * *   * * G * * * * *   * G G * * * * *   * G G G * * * * * * * * * * * </pre>
--	--

<pre> * * * * * * * * * * * P P P P P P P P P * * P * * * * * * * P * * P * * P * * * * * * * P * 25. * P P P P P P P P P * * P * * * * * * * * * * P * * P * * P * * * * * * * * * * * </pre>	<pre> * * * * * * * * * * * Q Q Q Q Q Q Q Q Q * * Q Q * * * * * * Q * * Q * Q * * Q * * Q * * Q * * Q * * Q * * Q * * Q * * * * * * Q Q * * Q Q Q Q Q Q Q Q Q * * * * * * * * * * * </pre>
--	--

```

* * * * *
* R R R R R R R R R *
* R * * * * * R * *
* R * * * * R * *
* R * * * * R * *
27. * R R R R R * *
* R * * * * R * *
* R * * * * R * *
* R * * * * * R * *
* R R R R R R R R R *
* * * * *

```

```

* * * * *
* K * * * * K *
* K * * * * K * *
* K * * * * K * *
* K * * * * K * *
28. * K K K K K * *
* K * * * * K * *
* K * * * * K * *
* K * * * * K * *
* K * * * * K * *
* * * * *

```

```

* * * * *
* J * * * * J * *
* * J * * * J * *
* * J * * * J * *
* * J * * * J * *
29. * * * * J * * *
* J * * * * J * *
* * J * * * J * *
* * J * * * J * *
* * J * * * J * *
* * * * *

```

```

* * * * *
* * * * *
* * N * * * *
* * N N * * *
* * N N N * *
30. * * * * * * * *
* * * * *
* * O * * * *
* * O O * * *
* * O O O * *
* * * * *

```

**Пример 6.** Дан вид квадратной символьной матрицы с числом строк и столбцов  $N = 11$  :

```

* * * * * * * * * * *
* * @ @ @ *           *
* $ * @ @ *           *
* $ $ * @ *           *
* $ $ $ * *           *
* * * * * * * * * * *
*           * * # # # *
*           * & * # # *
*           * & & * # *
*           * & & & * *
* * * * * * * * * * *

```

Требуется составить программу для ЭВМ, позволяющую получить на экране видеотерминала такую матрицу при значениях  $N = 1 - 20$  . Искомый результат может быть достигнут последовательным перебором элементов матрицы по строкам и столбцам на основе счетной циклической алгоритмической структуры. Из-за нерационального использования оперативной памяти нецелесообразно применение соответствующего одной строке одномерного символьного массива, равно как и соответствующего всей матрице двумерного массива. Достаточным является использование простых переменных (или констант), а формирование строк провести на основе форматного ввода-вывода. Обработка элементов матрицы в соответствии с требованиями задания должна составить тело вложенной циклической структуры.

```

C      ЗАДАНИЕ 6. Пример 6. Прогр. на языке FORTRAN.
      PROGRAM T6
C
      CHARACTER*1 C, C1, C2, C3, C4, C5, C6
      DATA C1, C2, C3, C4, C5, C6
      * / '*', '@', '#', '$', '&', ' ' /
C
      5 WRITE(*,10)
      10 FORMAT(1X,'Размер картинки [1-20]: N = ', $)
      READ(*,*) N
      IF( N .LT. 1 .OR. N .GT. 20 )          GOTO 5
C
      M = ( N + 1 ) / 2
      DO 95 I = 1 , N          ! Цикл по строкам
        DO 75 J = 1 , N      ! Цикл по столбцам
          C = C1
          IF( I .EQ. 1 .OR. I .EQ. N .OR. I .EQ. M
* .OR. J .EQ. 1 .OR. J .EQ. N .OR. J .EQ. M
* .OR. I .EQ. J )          GOTO 55
          IF( I .GT. J )          GOTO 15
          C = C2
          IF( J .LT. M )          GOTO 55
          C = C3
          IF( I .GT. M )          GOTO 55
          GOTO 35
      15      C = C4
          IF( I .LT. M )          GOTO 55
          C = C5
          IF( J .GT. M )          GOTO 55
      35      C = C6
C

```

```

55     WRITE(*,60)  C
60     FORMAT(A2,$)
75     CONTINUE      ! Конец цикла по столбцам
        WRITE(*,*)
C
95     CONTINUE      ! Конец цикла по строкам
        GOTO 5
        END

```

/\* ЗАДАНИЕ 6. Пример 6. Программа на языке C. \*/

```

#include <stdio.h>
#include <math.h>

int main( void ) {

char c;
char c1='*', c2='@', c3='#', c4='$', c5='&', c6=' ';
int i, j, k, m, n;

1: printf( " Размер картинки [1-20]: N = " );
   scanf ( "\f %d", &n );
   if( n < 1 || n > 20 ) goto 1;

   k = n + 1;
   m = k / 2;
   for( i = 1; i < k; i++ ) { /* Цикл по строкам */
     for( j = 1; j < k; j++ ){ /* Цикл по столбцам*/
       if( i == 1 || i == n || i == m || i == j
          || j == 1 || j == n || j == m ) c = c1;

```

```

else {
    if( i < j ) {
        if( j < m )      c = c2;
        else {
            if( i > m )   c = c3;
            else          c = c6;
        }
    }
    else {
        if( i < m )      c = c4;
        else {
            if( j > m )   c = c5;
            else          c = c6;
        }
    }
}
printf( "%2c", c );
} /* Конец цикла по столбцам */
printf( "\n" );
} /* Конец цикла по строкам */
goto l;
}

```

Размер картинки [1-20]: N = 7

```
* * * * * * *
* * @ *      *
* $ * *      *
* * * * * * *
*      * * # *
*      * & * *
* * * * * * *
```

Размер картинки [1-20]: N = 15

```
* * * * * * * * * * * * * * * *
* * @ @ @ @ @ *      *
* $ * @ @ @ @ *      *
* $ $ * @ @ @ *      *
* $ $ $ * @ @ *      *
* $ $ $ $ * @ *      *
* $ $ $ $ $ * *      *
* * * * * * * * * * * * * * * *
*      * * # # # # # *
*      * & * # # # # *
*      * & & * # # # *
*      * & & & * # # *
*      * & & & & * # *
*      * & & & & * *
* * * * * * * * * * * * * * *
```



## ЗАДАНИЕ 7.1

На основе принципа модульного программирования и применения пользовательских подпрограмм и функций составить программу для ЭВМ, позволяющую решить задачи аналитической геометрии на плоскости, определяемые пунктами **а)**, **б)**, **с)** вариантов задания. Вывести на экран видеотерминала и объяснить полученные результаты.

**Указание.** Должна быть учтена возможность правильной работы программы при произвольных численных значениях исходных данных вариантов задания.

### Варианты задания

1. Даны общие уравнения прямых:

$$3x + 2y - 1 = 0$$

$$7x - 10y + 12 = 0$$

$$11x + 4y - 9 = 0$$

$$5x - 8y + 6 = 0$$

Найти: **а)** уравнения с угловым коэффициентом заданных прямых; **б)** координаты точек пересечения заданных прямых с осями координат; **с)** площади треугольников, образованных заданными прямыми и осями координат.

2. Даны уравнения прямых с угловым коэффициентом:

$$y = -2x + 11$$

$$y = 3x + 8$$

$$y = -4x - 12$$

$$y = x - 7$$

Найти: **а)** общие уравнения заданных прямых; **б)** координаты точек пересечения заданных прямых с перпендикуля-

рами, опущенными из начала координат; **с**) площади треугольников, образованных тройками заданных прямых.

**3.** Даны координаты пар точек:

$( 5, 11 )$  и  $( -14, 2 )$  ;

$( -10, 4 )$  и  $( -13, -1 )$  ;

$( -7, -12 )$  и  $( 8, -6 )$  ;

$( 15, -9 )$  и  $( 3, 16 )$  .

Найти: **а**) нормальные уравнения прямых, проходящих через пары заданных точек; **б**) уравнения в отрезках прямых, проходящих через точку с координатами  $( 1, 4 )$  и параллельных прямой, проходящим через пары заданных точек; **с**) площади треугольников, образованных прямыми, проходящими через пары заданных точек и осями координат.

**4.** Даны координаты вершин четырехугольника:

$( 6, 5 )$ ;  $( -3, 7 )$ ;  $( -4, -2 )$ ;  $( 8, -1 )$  .

Найти: **а**) общие уравнения прямых, проходящих через вершины четырехугольника и параллельных прямой  $y = -x$  ; **б**) площади всех треугольников, вершинами которых являются вершины четырехугольника; **с**) координаты точки пересечения диагоналей четырехугольника.

**5.** Даны координаты вершин треугольника:

$( 8, 11 )$ ;  $( -13, 7 )$ ;  $( 4, -6 )$  .

Найти: **а**) внутренние углы треугольника; **б**) координаты точек пересечения биссектрис внутренних углов треугольника с его сторонами; **с**) нормальные уравнения медиан треугольника.

**6.** Даны общие уравнения прямых в соответствии с вариантом **1**. Найти: **а**) острые углы, образованные заданными прямыми и осями координат; **б**) координаты точек пересечения

чения прямой  $y = x$  и заданных прямых; **с**) длины сторон треугольников, образованных тройками заданных прямых.

**7.** Даны уравнения прямых с угловым коэффициентом в соответствии с вариантом **2**. Найти: **а**) нормальные уравнения заданных прямых; **б**) координаты точек пересечения прямой  $x + 2y + 4 = 0$  с заданными прямыми; **с**) расстояния от точки с координатами  $(3, 2)$  до центров тяжести треугольников, образованных тройками заданных прямых.

**8.** Даны координаты пар точек в соответствии с вариантом **3**. Найти: **а**) уравнения в отрезках прямых, проходящих через пары заданных точек; **б**) расстояния от точки с координатами  $(1, 3)$  до прямых, проходящих через пары заданных точек; **с**) координаты центров тяжести треугольников, образованных прямыми, проходящими через пары заданных точек и осями координат.

**9.** Даны координаты вершин четырехугольника в соответствии с вариантом **4**. Найти: **а**) нормальные уравнения прямых, совпадающих со сторонами четырехугольника; **б**) площади всех треугольников, образованных диагоналями четырехугольника и одной из его сторон; **с**) координаты точек пересечения диагоналей четырехугольника с осями.

**10.** Даны координаты вершин треугольника в соответствии с вариантом **5**. Найти: **а**) острые углы пересечения медиан со сторонами треугольника; **б**) расстояния от точки с координатами  $(3, 3)$  до прямых, совпадающих со сторонами треугольника; **с**) уравнения в отрезках биссектрис треугольника.

**11.** Даны общие уравнения прямых в соответствии с вариантом **1**. Найти: **а**) уравнения заданных прямых в отрез-

ках; **б)** координаты точек пересечения заданных прямых попарно между собой; **с)** длины сторон треугольников, образованных заданными прямыми и осями координат.

**12.** Даны уравнения прямых с угловым коэффициентом в соответствии с вариантом **2**. Найти: **а)** общие уравнения прямых, проходящих через начало координат и перпендикулярных заданным прямым; **б)** расстояния от начала координат до точек пересечения прямой  $y = 7x - 6$  с заданными прямыми; **с)** внутренние углы треугольников, образованных тройками заданных прямых.

**13.** Даны координаты пар точек в соответствии с вариантом **3**. Найти: **а)** уравнения с угловым коэффициентом прямых, проходящих через пары заданных точек; **б)** общие уравнения прямых, проходящих через точку с координатами  $(4, 1)$  и параллельных прямым, проходящим через пары заданных точек; **с)** координаты центров тяжести треугольников, образованных тройками прямых, проходящих через пары заданных точек.

**14.** Даны координаты вершин четырехугольника в соответствии с вариантом **4**. Найти: **а)** уравнения в отрезках прямых, совпадающих со сторонами четырехугольника; **б)** расстояния от вершин четырехугольника до прямой  $2x - 3y + 1 = 0$ ; **с)** координаты точек пересечения диагоналей четырехугольника с биссектрисами его противоположных внутренних углов.

**15.** Даны координаты вершин треугольника в соответствии с вариантом **5**. Найти: **а)** острые углы пересечения биссектрис внутренних углов со сторонами треугольника; **б)** расстояния от центра тяжести до сторон треугольника;

с) уравнения с угловым коэффициентом высот треугольника.

**16.** Даны общие уравнения прямых в соответствии с вариантом **1**. Найти: **а)** внутренние углы треугольников, образованных тройками заданных прямых; **б)** координаты точек пересечения заданных прямых и прямой  $3x - 2y + 4 = 0$ ; **с)** площади треугольников, образованных тройками заданных прямых.

**17.** Даны уравнения прямых с угловым коэффициентом в соответствии с вариантом **2**. Найти: **а)** уравнения заданных прямых в отрезках; **б)** координаты точек пересечения заданных прямых попарно между собой; **с)** длины сторон треугольников, образованных заданными прямыми и осями координат.

**18.** Даны координаты пар точек в соответствии с вариантом **3**. Найти: **а)** общие уравнения прямых, проходящих через пары заданных точек; **б)** расстояния от точки с координатами  $(2, 1)$  до прямых, проходящих через пары заданных точек; **с)** площади треугольников, образованных тройками прямых, проходящих через пары заданных точек.

**19.** Даны координаты вершин четырехугольника в соответствии с вариантом **4**. Найти: **а)** уравнения с угловым коэффициентом прямых, совпадающих со сторонами четырехугольника; **б)** координаты центров тяжести всех треугольников, вершинами которых являются вершины четырехугольника; **с)** соотношения, в которых диагонали четырехугольника делятся своей точкой пересечения.

**20.** Даны координаты вершин треугольника в соответствии с вариантом **5**. Найти: **а)** уравнения с угловым ко-

эффицентом сторон треугольника; **б**) координаты середин сторон треугольника; **с**) расстояния от начала координат до прямых, проходящих через вершины треугольника и параллельных их противолежащим сторонам.

**21.** Даны общие уравнения прямых в соответствии с вариантом **1**. Найти: **а**) нормальные уравнения заданных прямых; **б**) расстояния от точки с координатами  $(1, 1)$  до заданных прямых; **с**) координаты центров тяжести треугольников, образованных заданными прямыми и осями координат.

**22.** Даны уравнения прямых с угловым коэффициентом в соответствии с вариантом **2**. Найти: **а**) общие уравнения прямых, проходящих через точку с координатами  $(4, -3)$  и перпендикулярных заданным прямым; **б**) координаты точек пересечения прямой  $3x - y + 5 = 0$  и заданных прямых; **с**) длины медиан треугольников, образованных тройками заданных прямых.

**23.** Даны координаты пар точек в соответствии с вариантом **3**. Найти: **а**) острые углы, образуемые попарным пересечением прямых, проходящих через пары заданных точек; **б**) нормальные уравнения прямых, проходящих через точку с координатами  $(5, 1)$  и параллельных прямым, проходящим через пары заданных точек; **с**) расстояния от точки с координатами  $(-4, 5)$  до центров тяжести треугольников, образованных тройками прямых, проходящих через пары заданных точек.

**24.** Даны координаты вершин четырехугольника в соответствии с вариантом **4**. Найти: **а**) внутренние углы четырехугольника; **б**) расстояния от точки с координатами  $(5, -6)$  до центров тяжести всех треугольников, вершина-

ми которых являются вершины четырехугольника; **с**) соотношения, в которых диагонали четырехугольника делятся биссектрисами его внутренних углов.

**25.** Даны координаты вершин треугольника в соответствии с вариантом **5**. Найти: **а**) общие уравнения сторон треугольника; **б**) соотношения, в которых биссектрисы внутренних углов треугольника делят его стороны; **с**) расстояния от точки с координатами  $(-3, -3)$  до прямых, совпадающих с высотами треугольника.

**26.** Даны общие уравнения прямых в соответствии с вариантом **1**. Найти: **а**) острые углы, образованные заданными прямыми и прямой  $y = x$ ; **б**) расстояния от точки с координатами  $(-1, -1)$  до заданных прямых; **с**) расстояния от начала координат до центров тяжести треугольников, образованных тройками заданных прямых.

**27.** Даны уравнения прямых с угловым коэффициентом в соответствии с вариантом **2**. Найти: **а**) нормальные уравнения прямых, проходящих через точку с координатами  $(-2, -3)$  и параллельных заданным прямым; **б**) координаты точек пересечения заданных прямых и прямой  $3x - 2y + 4 = 0$ ; **с**) расстояния от начала координат до центров тяжести треугольников, образованных тройками заданных прямых.

**28.** Даны координаты пар точек в соответствии с вариантом **3**. Найти: **а**) координаты точек, образуемых попарным пересечением прямых, проходящих через пары заданных точек; **б**) уравнения с угловым коэффициентом прямых, проходящих через точку с координатами  $(1, 5)$  и параллельных прямым, проходящим через пары заданных точек; **с**) расто-

яния от точки с координатами  $(4, -5)$  до диагоналей прямоугольника, образованного прямыми, проходящими через пары заданных точек.

**29.** Даны координаты вершин четырехугольника в соответствии с вариантом **4**. Найти: **а)** длины сторон и диагоналей четырехугольника; **б)** расстояния от точки с координатами  $(-5, 6)$  до вершин четырехугольника; **с)** общие уравнения медиан всех треугольников, вершинами которых являются вершины четырехугольника.

**30.** Даны координаты вершин треугольника в соответствии с вариантом **5**. Найти: **а)** острые углы между медианами и биссектрисами, проходящими через одни и те же вершины треугольника; **б)** общие уравнения медиан треугольника; **с)** расстояния от вершин треугольника до прямой  $y = x - 3$ .

**Пример 7.1.** Даны координаты точек на плоскости:  $(1, 1)$ ;  $(-1, 1)$ ;  $(-1, -1)$ ;  $(1, -1)$ .

Найти: **а)** координаты точек тяжести всех треугольников, вершинами которых являются заданные точки; **б)** общие уравнения всех прямых, проходящих через пары заданных точек; **с)** расстояния от начала координат до полученных прямых.

**С** ЗАДАНИЕ 7.1. Пример 7.1. Progr. на яз. FORTRAN.  
PROGRAM T71

**С**  
LOGICAL LL  
COMMON / XY / X(4), Y(4)  
DATA X / 1., -1., -1., 1. /  
DATA Y / 1., 1., -1., -1. /



```

C
C   WC - функция расч. координаты центра тяжести
C   WC( W1, W2, W3 ) = 0.33333333*(W1 + W2 + W3)
C
N = 4          ! Количество заданных точек
DO 40 I = 1 , N ! Цикл по заданным точкам
  DO 30 J = I+1 , N
    DO 10 K = J+1 , N
      XC = WC( X(I), X(J), X(K) )
      YC = WC( Y(I), Y(J), Y(K) )
      WRITE(*,50)
      WRITE(*,60) I, J, K, XC, YC
10    CONTINUE
      CALL ABC( I, J, A, B, C, LL )
      IF( LL )          GOTO 20
      WRITE(*,50)
      WRITE(*,70) I, J
      GOTO 30
20    WRITE(*,50)
      WRITE(*,80) I, J, A, B, C
      WRITE(*,90) R ( A, B, C )
30    CONTINUE
40  CONTINUE      ! Конец цикла по заданным точкам
C
50  FORMAT(' Точки:', $)
60  FORMAT(3I2, '; Xc, Yc :', 2E10.3)
70  FORMAT(2I2, ' совпадают ...')
80  FORMAT(2I2, '; A, B, C :', 3E10.3)
90  FORMAT(19X, 'R :', E10.3)
STOP
END

```

```

C
C   Модуль расч. расстояния от прямой до т. (0,0)
FUNCTION R( A, B, C )
Q = SQRT( A**2 + B**2 )
IF( C .LT. 0. ) Q = -Q
R = C / Q
RETURN
END

C
C   Модуль расчета коэф-тов общего уравн. прямой
SUBROUTINE ABC( I, J, A, B, C, LL )
LOGICAL LL, LX, LY
COMMON / XY / X(4), Y(4)
A = 1.0
B = 1.0
LL = .F A L S E.
LX = ABS( X(I) - X(J) ) .LT. 1.E-7
LY = ABS( Y(I) - Y(J) ) .LT. 1.E-7
IF( LX .AND. LY ) R E T U R N
LL = .T R U E.
IF( .NOT. LX ) GOTO 30
  B = 0.0
  C = -X(I)
  R E T U R N
30 IF( .NOT. LY ) GOTO 60
  A = 0.0
  C = -Y(I)
  R E T U R N
60 D = Y(I) - Y(J)
B = ( X(J) - X(I) ) / D
C = ( X(I)*Y(J) - X(J)*Y(I) ) / D

```

R E T U R N  
E N D

```
Точки: 1 2 3; Xc,Yc : -.333E+00 .333E+00
Точки: 1 2 4; Xc,Yc : .333E+00 .333E+00
Точки: 1 2; A, B, C : .000E+00 .100E+01 -.100E+01
      R : .100E+01
Точки: 1 3 4; Xc,Yc : .333E+00 -.333E+00
Точки: 1 3; A, B, C : .100E+01 -.100E+01 .000E+00
      R : .000E+00
Точки: 1 4; A, B, C : .100E+01 .000E+00 -.100E+01
      R : .100E+01
Точки: 2 3 4; Xc,Yc : -.333E+00 -.333E+00
Точки: 2 3; A, B, C : .100E+01 .000E+00 .100E+01
      R : .100E+01
Точки: 2 4; A, B, C : .100E+01 .100E+01 .000E+00
      R : .000E+00
Точки: 3 4; A, B, C : .000E+00 .100E+01 .100E+01
      R : .100E+01
```

/\* ЗАДАНИЕ 7.1. Пример 7.1. Программа на языке C.\*/

```
#include <stdio.h>
#include <math.h>

double x[] = { 1., -1., -1., 1. };
double y[] = { 1., 1., -1., -1. };
double r( double, double, double );
double wc( double, double, double );
int abc( int,int, double *, double *, double *);
```

```

int main( void ) {
int i, j, k;

int n = 4; /* Количество заданных точек */
double a, b, c;

for( i = 0; i < n; i++ ) { /* Цикл по точкам */
for( j = i+1; j < n; j++ ) {
for( k = j+1; k < n; k++ ) {
printf( "Точки:" );
printf("%2d%2d%2d; Xc,Yc :%10.2e%10.2e\n",
i+1, j+1, k+1, wc( x[i], x[j], x[k] ),
wc( y[i], y[j], y[k] ));
}
printf( "Точки:" );
if( abc( i, j, &a, &b, &c ) ) {
printf("%2d%2d; A, B, C :", i+1, j+1 );
printf("%10.2e%10.2e%10.2e\n",a, b, c );
printf( " R :%10.2e\n",
r( a, b, c ) );
}
else {
printf("%2d%2d совпадают ... \n", i+1,j+1 );
}
}
} /* Конец цикла по заданным точкам */
return 1;
}

```

```

/* Функция расчета координаты центра тяжести */
double wc( double w1, double w2, double w3 ) {
    return 0.3333333333333333 * ( w1 + w2 + w3 );
}

```

```

/* Функция расчета раст. от прямой до т. (0,0)*/
double r( double a, double b, double c ) {
    double q;
    q = sqrt( pow( a, 2. ) + pow( b, 2. ) );
    if( c < 0. ) q = -q;
    return c / q;
}

```

```

/* Функция расчета коэф. общего уравн. прямой */
int abc( int i, int j,
         double *a, double *b, double *c ) {
    int lx, ly;
    double d;
    *a = 1.0;
    *b = 1.0;
    lx = fabs( x[i] - x[j] ) < 1.e-16;
    ly = fabs( y[i] - y[j] ) < 1.e-16;
    if( lx && ly ) return 0;
    if( lx ) {
        *b = 0.0;
        *c = -x[i];
        return 1;
    }
    else if( ly ) {
        *a = 0.0;
        *c = -y[i];
    }
}

```

```

    return 1;
}
d = y[i] - y[j];
*b = ( x[j] - x[i] ) / d;
*c = ( x[i]*y[j] - x[j]*y[i] ) / d;
return 1;
}

```

```

Точки: 1 2 3; Xc,Yc : -3.33e-01  3.33e-01
Точки: 1 2 4; Xc,Yc :  3.33e-01  3.33e-01
Точки: 1 2; A, B, C :  0.00e+00  1.00e+00 -1.00e+00
                R :  1.00e+00
Точки: 1 3 4; Xc,Yc :  3.33e-01 -3.33e-01
Точки: 1 3; A, B, C :  1.00e+00 -1.00e+00  0.00e+00
                R :  0.00e+00
Точки: 1 4; A, B, C :  1.00e+00  0.00e+00 -1.00e+00
                R :  1.00e+00
Точки: 2 3 4; Xc,Yc : -3.33e-01 -3.33e-01
Точки: 2 3; A, B, C :  1.00e+00  0.00e+00  1.00e+00
                R :  1.00e+00
Точки: 2 4; A, B, C :  1.00e+00  1.00e+00  0.00e+00
                R :  0.00e+00
Точки: 3 4; A, B, C :  0.00e+00  1.00e+00  1.00e+00
                R :  1.00e+00

```

## ЗАДАНИЕ 7.2

На основе принципа работы и применения пользовательских подпрограмм и функций, составить программу для ЭВМ, позволяющую решить задачи, определяемые условиями вариантов задания. Вывести на экран видеотерминала и объяснить полученные результаты.

### Варианты задания

1. В основной программе описана переменная типа **REAL\*8 R8** и ей присвоено значение **1.D0** .

В модуле **SUBROUTINE SUBR( W8 )** описана переменная типа **COMPLEX\*8 W8** и выполняется операция **WRITE(\*,\*) W8** .

Имеет место вызов **CALL SUBR( R8 )** .

Какое значение появится на экране после этого вызова ?

2. В основной программе описан массив типа **INTEGER\*1 I1(100)** и каждому элементу этого массива присвоено значение **1** .

В модуле **SUBROUTINE SUBR( K1 )** описан массив типа **INTEGER\*1 K1(20)** и выполняется цикл

$$K1(I) = I, \quad I = 1 - 20 .$$

Какие значения будут иметь элементы массива **I1** в результате последовательных вызовов

**CALL SUBR( I1(11) ) , CALL SUBR( I1(21) ) ?**

3. В основной программе описана переменная типа **REAL\*8 R8** и ей присвоено произвольное значение.

Написать функцию, возвращающую логическое значение **.TRUE.** , если значение **R8** положительно, и логическое значение **.FALSE.** в противном случае.

4. В основной программе описана переменная типа **INTEGER\*4 I4** и ей присвоено значение **4** .

В модуле **SUBROUTINE SUBR( R4 )** описана переменная типа **REAL\*4 R4** и выполняется операция

**WRITE(\*,\*) R4 .**

Имеет место вызов **CALL SUBR( I4 )** .

Какое значение появится на экране после этого вызова ?

5. В основной программе описан массив типа **REAL\*4 R4(100)** и каждому элементу этого массива присвоено значение **1.0** .

В модуле **SUBROUTINE SUBR( V4 )** описан массив типа **REAL\*4 V4(40)** и выполняется цикл

**V4(I) = FLOAT( I ) , I = 1 - 40 .**

Какие значения будут иметь элементы массива **R4** в результате последовательных вызовов

**CALL SUBR( R4(21) ) , CALL SUBR( R4(31) ) ?**

6. В основной программе описана переменная типа **INTEGER\*4 I4** и ей присвоено произвольное значение.

Написать функцию, возвращающую текстовое значение **'GE. 0'** , если значение **I4** неотрицательно, и текстовое значение **'LT. 0'** в противном случае.

7. В основной программе описан массив типа **INTEGER\*1 I1(100)** и каждому элементу этого массива присвоено значение **1** .

В модуле **SUBROUTINE SUBR( L1 )** описан массив типа **LOGICAL\*1 L1(20)** и выполняется цикл

**L1(I) = .FALSE. , I = 1 - 20 .**

Какие значения будут иметь элементы массива **I1** в результате последовательных вызовов

**CALL SUBR( I1(21) ) , CALL SUBR( I1(41) ) ?**



8. В основной программе описана переменная типа **CHARACTER\*8 C8** и ей присвоено текстовое значение **'ABCDEFGH'**.

В модуле **SUBROUTINE SUBR( R8 )** описана переменная типа **REAL\*8 R8** и выполняется операция

**WRITE(\*,\*) R8 .**

Имеет место вызов **CALL SUBR( C8 )**.

Какое значение появится на экране после этого вызова ?

9. В основной программе описана переменная типа **LOGICAL LL** и ей присвоено произвольное значение.

Написать функцию, возвращающую текстовые значения **'TRUE.'** или **'FALSE.'** в зависимости от значения **LL**.

10. В основной программе описан массив типа **INTEGER\*1 I1(100)** и каждому элементу этого массива присвоено значение **1**.

В модуле **SUBROUTINE SUBR( C1 )** описан массив типа **CHARACTER\*1 C1(10)** и выполняется цикл

**C1(I) = 'A' , I = 1 - 10 .**

Какие значения будут иметь элементы массива **I1** в результате последовательных вызовов

**CALL SUBR( I1(31) ) , CALL SUBR( I1(36) ) ?**

11. В основной программе описана переменная типа **INTEGER\*4 I4** и ей присвоено значение **1**.

В модуле **SUBROUTINE SUBR( L4 )** описана переменная типа **LOGICAL\*4 L4** и выполняется операция

**WRITE(\*,'(L2)') L4 .**

Имеет место вызов **CALL SUBR( I4 )**.

Какое значение появится на экране после этого вызова ?

12. В основной программе описана переменная типа **CHARACTER CС** и ей присвоено произвольное значение.

Написать функцию, возвращающую значение **'Y E S'**, если значение **СС** совпадает с **'@'**, и значение **'N O'** в противном случае.

13. В основной программе описан массив типа **INTEGER\*2 I2(100)** и каждому элементу этого массива присвоено значение **10**.

В модуле **SUBROUTINE SUBR( K2 )** описан массив типа **INTEGER\*2 K2(10)** и выполняется цикл

$$K2(I) = I * 10, I = 1 - 10.$$

Какие значения будут иметь элементы массива **I2** в результате последовательных вызовов

**CALL SUBR( I2(41) ), CALL SUBR( I2(91) ) ?**

14. В основной программе описана переменная типа **REAL\*4 R4** и ей присвоено произвольное значение.

Написать функцию, возвращающую текстовое значение **'GE. 0.0'**, если значение **R4** неотрицательно, и текстовое значение **'LT. 0.0'** в противном случае.

15. В основной программе описана переменная типа **CHARACTER\*4 C4** и ей присвоено текстовое значение **'ABCD'**.

В модуле **SUBROUTINE SUBR( R4 )** описана переменная типа **REAL\*4 R4** и выполняется операция

**WRITE(\*,\*) R4**.

Имеет место вызов **CALL SUBR( C4 )**.

Какое значение появится на экране после этого вызова ?

16. В основной программе описан массив типа **REAL\*8 R8(100)** и каждому элементу этого массива

ва присвоено значение **1.D0** .

В модуле **SUBROUTINE SUBR( V8 )** описан массив типа **REAL\*8 V8(40)** и выполняется цикл

**V8(I) = DFLOAT( I ) , I = 1 - 40 .**

Какие значения будут иметь элементы массива **R8** в результате последовательных вызовов

**CALL SUBR( R8(16) ) , CALL SUBR( R8(56) ) ?**

**17.** В основной программе описана переменная типа **REAL\*4 R4** и ей присвоено значение **0.0** .

В модуле **SUBROUTINE SUBR( L4 )** описана переменная типа **LOGICAL\*4 L4** и выполняется операция

**WRITE(\*,'(L2)') L4 .**

Имеет место вызов **CALL SUBR( R4 )** .

Какое значение появится на экране после этого вызова ?

**18.** В основной программе описан массив типа **LOGICAL\*1 L1(100)** и каждому элементу этого массива присвоено логическое значение **.TRUE.** .

В модуле **SUBROUTINE SUBR( I1 )** описан массив типа **INTEGER\*1 I1(30)** и выполняется цикл

**I1(I) = 0 , I = 1 - 30 .**

Какие значения будут иметь элементы массива **L1** в результате последовательных вызовов

**CALL SUBR( L1(26) ) , CALL SUBR( L1(66) ) ?**

**19.** В основной программе описана переменная типа **INTEGER\*4 I4** и ей присвоено произвольное значение.

Написать функцию, возвращающую логическое значение **.TRUE.** , если значение **I4** положительно, и логическое значение **.FALSE.** в противном случае.

**20.** В основной программе описан массив типа `CHARACTER*1 C1(100)` и каждому элементу этого массива присвоено значение `'A'`.

В модуле `SUBROUTINE SUBR( I1 )` описан массив типа `INTEGER*1 I1(26)` и выполняется цикл

$$I1(I) = 64 + I, \quad I = 1 - 26.$$

Какие значения будут иметь элементы массива `C1` в результате последовательных вызовов

`CALL SUBR( C1(31) ), CALL SUBR( C1(51) ) ?`

**21.** В основной программе описана переменная типа `CHARACTER*8 C8` и ей присвоено значение `'ABCDEFGH'`.

В модуле `SUBROUTINE SUBR( W8 )` описана переменная типа `COMPLEX*8 W8` и выполняется операция `WRITE(*,*) W8`.

Имеет место вызов `CALL SUBR( C8 )`.

Какое значение появится на экране после этого вызова ?

**22.** В основной программе описана переменная типа `LOGICAL*4 L4` и ей присвоено логическое значение `.TRUE.`.

В модуле `SUBROUTINE SUBR( I4 )` описана переменная типа `INTEGER*4 I4` и выполняется операция

`WRITE(*,*) ACOS( -FLOAT( I4 ) )`.

Имеет место вызов `CALL SUBR( L4 )`.

Какое значение появится на экране после этого вызова ?

**23.** В основной программе описан массив типа `INTEGER*4 I4(100)` и каждому элементу этого массива присвоено значение `1000`.

В модуле `SUBROUTINE SUBR( K4 )` описан массив типа `INTEGER*4 K4(30)` и выполняется цикл

$$K4(I) = I * 1000, I = 1 - 30.$$

Какие значения будут иметь элементы массива **I4** в результате последовательных вызовов

**CALL SUBR( I4(11) ), CALL SUBR( I4(26) ) ?**

**24.** В основной программе описана переменная типа **REAL\*8 R8** и ей присвоено произвольное значение.

Написать функцию, возвращающую текстовое значение **'GE. 0.D0'**, если значение **R8** неотрицательно, и текстовое значение **'LT. 0.D0'** в противном случае.

**25.** В основной программе описана переменная типа **REAL\*4 R4** и ей присвоено значение **4.0**.

В модуле **SUBROUTINE SUBR( I4 )** описана переменная типа **INTEGER\*4 I4** и выполняется операция

**WRITE(\*,\*) I4 .**

Имеет место вызов **CALL SUBR( R4 )**.

Какое значение появится на экране после этого вызова ?

**26.** В основной программе описана переменная типа **CHARACTER\*2 C2** и ей присвоено текстовое значение **'10'**.

В модуле **SUBROUTINE SUBR( I2 )** описана переменная типа **INTEGER\*2 I2** и выполняется операция

**WRITE(\*,\*) I2 .**

Имеет место вызов **CALL SUBR( C2 )**.

Какое значение появится на экране после этого вызова ?

**27.** В основной программе описана переменная типа **LOGICAL\*4 L4** и ей присвоено значение **.FALSE.**

В модуле **SUBROUTINE SUBR( R4 )** описана переменная типа **REAL\*4 R4** и выполняется операция

**WRITE(\*,\*) COS( R4 ) .**

Имеет место вызов `CALL SUBR( L4 )` .

Какое значение появится на экране после этого вызова ?

**28.** В основной программе описана переменная типа `CHARACTER*4 C4` и ей присвоено значение `'ABCD'` . В модуле `SUBROUTINE SUBR( I4 )` описана переменная типа `INTEGER*4 I4` и выполняется операция `WRITE(*,*) I4` .

Имеет место вызов `CALL SUBR( C4 )` .

Какое значение появится на экране после этого вызова ?

**29.** В основной программе описана переменная типа `REAL*4 R4` и ей присвоено произвольное значение.

Написать функцию, возвращающую логическое значение `.TRUE.` , если значение `R4` положительно, и логическое значение `.FALSE.` в противном случае.

**30.** В основной программе описана переменная типа `CHARACTER*16 C16` и ей присвоено текстовое значение `' A B C D E F G H'` .

В модуле `SUBROUTINE SUBR( W16 )` описана переменная типа `COMPLEX*16 W16` и выполняется операция `WRITE(*,*) W16` .

Имеет место вызов `CALL SUBR( C16 )` .

Какое значение появится на экране после этого вызова ?

**Пример 7.2а.** В косоугольном непротиворечивом треугольнике, где `a, b, c` - длины сторон, а  `$\alpha, \beta, \gamma$`  - противолежащие этим сторонам углы, по теореме косинусов:

$$a^2 = b^2 + c^2 - 2bc \cos \alpha ;$$

$$b^2 = a^2 + c^2 - 2ac \cos \beta ;$$

$$c^2 = a^2 + b^2 - 2ab \cos \gamma .$$

Найти значения внутренних углов треугольника, зная длины его сторон.

```
C      ЗАДАНИЕ 7.2а.Пример 7.2а.Прогр.на яз.FORTRAN.
C      Оператор - функция.
C
      PROGRAM T72A
*
      R E A L*8  FNC, A, B, C, ALP, BET, GAM, Q,
*              SUM, X, Y, Z
*
      PARAMETER( Q = 1.D-38 )
*
      FNC( X, Y, Z ) = DACOS( ( Y*Y + Z*Z - X*X )
*                          / ( 2.DO * Y * Z ) )
*
11  WRITE(*,*)'A, B, C : '
    READ (*,*) A, B, C
*
    IF( A.LT.Q .OR. B.LT.Q .OR. C.LT.Q )GOTO 11
    IF( A+B .LT. C ) GOTO 11
    IF( A+C .LT. B ) GOTO 11 ! Треугольник
    IF( B+C .LT. A ) GOTO 11 ! противоречив
*
    ALP = FNC( A, B, C )
*
    BET = FNC( B, A, C )
*
    GAM = FNC( C, A, B )
*
    SUM = ALP + BET + GAM
```

```

*
WRITE(*,*) 'ALP, BET, GAM :'
WRITE(*,55) ALP, BET, GAM
WRITE(*,*) 'SUM :'
WRITE(*,55) SUM
55  FORMAT(3G17.10)
*
GOTO 11
*
END

A, B, C :
3. 4. 5.

ALP, BET, GAM :
.6435011088      .9272952180      1.570796327

SUM :
3.141592654

```

**Пример 7.2b.** При произвольных значениях параметра  $n$  и аргумента  $x$  вычислить выражение

$$y = ax^2 + bx + c ,$$

если входящие в него коэффициенты определяются через заранее заданные элементы массивов  $Q(100)$  и  $R(20)$  следующим образом:

$$a = \sum_{i=1}^n Q_i ; \quad b = \sum_{i=n+1}^{100} Q_i ; \quad c = \sum_{i=1}^{20} R_i .$$



Смысл задания заключается в необходимости организации многократного суммирования элементов разных одномерных массивов в произвольных пределах.

С ЗАДАНИЕ 7.2b.Пример 7.2b.Прогр.на яз.FORTRAN.  
С Процедура - функция.

```
С
PROGRAM T72B
*
REAL*8    A, B, C, SUM, X, Y, Q(100), R(20)
*
DATA      Q/ 40*1.D0, 60*2.D0 /, R/ 20*3.D0 /
*
11 WRITE(*,*) 'N, X :'
   READ (*,*) N, X
   IF( N .LT. 1 .OR. N .GT. 99 ) GOTO 11
*
   A = SUM( Q(1), N )
*
   B = SUM( Q(N+1), 100-N )
*
   C = SUM( R, 20 )
*
   Y = A*X**2 + B*X + C
*
   WRITE(*,*) 'A, B, C :'
   WRITE(*,55) A, B, C
   WRITE(*,*) 'Y :'
   WRITE(*,55) Y
55 FORMAT(3G10.4)
*
```

```

GOTO 11
END
*
REAL*8 FUNCTION SUM( Z, M )
*
REAL*8 Z(M)      ! S, Z(1)
*
SUM = 0.DO       ! S = 0.DO
*
DO I = 1, M
    SUM = SUM + Z(I) ! S = S + Z(I)
END DO
*
                                ! SUM = S
RETURN
END

```

N, X :  
40, 1.

A, B, C :  
40.00      120.0      60.00

Y :  
220.0

**Пример 7.2с.** Найти попарно суммы и скалярные произведения трех  $n$ -мерных векторов. Суммой векторов

$$\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) ; \quad \mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$$

является вектор с компонентами в виде сумм компонент исходных векторов

$$S = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n),$$

а их скалярным произведением является число

$$p = \sum_{i=1}^n a_i \cdot b_i.$$

C ЗАДАНИЕ 7.2с.Пример 7.2с.Прогр.на яз.FORTRAN.

C Подпрограмма общего вида.

C

```
PROGRAM T72C
```

\*

```
PARAMETER( M = 10 )
```

\*

```
REAL*8 A(M), B(M), C(M), P(3), S(M,3)
```

\*

```
DATA A, B, C/ M*1.D0, M*2.D0, M*3.D0 /
```

\*

```
11 WRITE(*,15)
```

```
15 FORMAT(' N = ', $ )
```

```
READ (*,*) N
```

```
IF( N .GT. M ) GOTO 11
```

\*

```
CALL SUBV( N, A, B, P(1), S(1,1) )
```

\*

```
CALL SUBV( N, A, C, P(2), S(1,2) )
```

\*

```
CALL SUBV( N, B, C, P(3), S(1,3) )
```

\*

```
DO J = 1, 3
```

```
WRITE(*,*)
```

```
WRITE(*,55) J, P(J)
```

```

        WRITE(*,95)  J, ( S(I,J), I = 1, N )
    END DO
*
    GOTO 11
*
55  FORMAT(' P(',I1,') = ',F4.1)
95  FORMAT(' S(',I1,') : ',10F4.1)
*
    END
*
*
*
    SUBROUTINE  SUBV( N, A, B, P, S )
*
    R E A L*8   A(N), B(N), P, S(N)
*
    P = 0.DO
*
    DO  I = 1, N
        P = P  +  A(I) * B(I)
        S(I)  =  A(I) + B(I)
    END DO
*
    RETURN
    END

N = 8

P(1) = 16.0
S(1) :  3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0

```

P(2) = 24.0  
S(2) : 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0

P(3) = 48.0  
S(3) : 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0

**Пример 7.2d.** В основной программе описана переменная типа **CHARACTER\*8 C8** и ей присвоено текстовое значение **'ABCDEFGH'**.

В модуле **SUBROUTINE SUBR1( I1 )** описан массив типа **INTEGER\*1 I1(8)** и выполняется операция **WRITE(\*,'(8I4)') I1**.

В модуле **SUBROUTINE SUBR2( I2 )** описан массив типа **INTEGER\*2 I2(4)** и выполняется операция **WRITE(\*,'(4I8)') I2**.

Какие значение появятся на экране в результате последовательных вызовов

**CALL SUBR1( C8 ) , CALL SUBR2( C8 ) ?**

```
C      ЗАДАНИЕ 7.2d.Пример 7.2d.Прогр.на яз.FORTRAN.  
      PROGRAM T72D  
*  
      CHARACTER*8 C8  
      C8 = 'ABCDEFGH'  
*  
      CALL SUBR1( C8 )  
*  
      CALL SUBR2( C8 )  
*  
      STOP  
      END
```

```

*
SUBROUTINE SUBR1( I1 )
INTEGER*1 I1(8)
*
WRITE(*,'(8I4)') I1
*
RETURN
END
*
SUBROUTINE SUBR2( I2 )
INTEGER*2 I2(4)
*
WRITE(*,'(4I8)') I2
*
RETURN
END

65 66 67 68 69 70 71 72
16961 17475 17989 18503

```

При вызовах подпрограмм **SUBR1** и **SUBR2**, в качестве фактического параметра передается адрес начального байта текстовой переменной **C8**, то есть символа **'A'**. При выводе на экран видеотерминала целочисленных значений элементов массивов **I1** и **I2** в соответствующих подпрограммах наблюдаются числа, определяемые кодировкой либо одного, либо двух последовательных байт текстовой переменной **C8**. В последнем случае все получаемые числа нечетны, что явно указывает на адресный (младший) байт целочисленного массива **I2**.

**Пример 7.2е.** В основной программе описан массив типа **CHARACTER C1( 26 ) .**

В модуле **SUBROUTINE SUBR1( I1 )** описан массив типа **INTEGER\*1 I1(26)** и выполняется цикл

$$I1(I) = I + 64 , I = 1 - 26 .$$

В модуле **SUBROUTINE SUBR2( I1 )** описан массив типа **INTEGER\*1 I1(6)** и выполняется цикл

$$I1(I) = 32 , I = 1 - 6 .$$

Какие значения будут иметь элементы массива **C1** в результате последовательных вызовов **CALL SUBR1(C1), CALL SUBR2( C1(5) ) , CALL SUBR2( C1(15) ) ?**

```
C      ЗАДАНИЕ 7.2е.Пример 7.2е.Прогр.на яз.FORTRAN.
      PROGRAM T72E
*
      CHARACTER    C1(26)
*
      CALL SUBR1( C1 )
      WRITE(*,55) C1
*
      CALL SUBR2( C1(5) )
      WRITE(*,55) C1
*
      CALL SUBR2( C1(15) )
      WRITE(*,55) C1
55  FORMAT(26A2)
*
      STOP
*
      END
*
```

```

SUBROUTINE SUBR1( I1 )
INTEGER*1 I1(26)
*
DO I = 1, 26
    I1(I) = I + 64
END DO
*
RETURN
*
END
*
SUBROUTINE SUBR2( I1 )
INTEGER*1 I1(6)
*
DO I = 1, 6
    I1(I) = 32
END DO
*
RETURN
*
END

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D           K L M N O P Q R S T U V W X Y Z
A B C D           K L M N           U V W X Y Z

```

При вызове подпрограммы **SUBR1**, в качестве фактического параметра передается адрес 1-го элемента массива **C1**. Начиная с этого адреса, эта подпрограмма модифицирует однобайтовые элементы массива **I1** целочисленными значениями, соответствующими кодировке 26 заглавных



букв латинского алфавита.

При вызовах подпрограммы **SUBR2** , в качестве фактических параметров передаются адреса соответственно 5-го и 15-го элементов массива **C1** . Начиная с этих адресов, эта подпрограмма модифицирует 6 однобайтовых элементов массива **I1** целочисленными значениями, соответствующими кодировке символа "пробел".

Имеено это наблюдается при следующих за вызовами подпрограмм выводами на экран видеотерминала значений элементов массива **C1** .

## ЗАДАНИЕ 8

Используя возможности форматного вывода, составить программу для ЭВМ, позволяющую сформировать таблицу, колонки которой содержат одномерные массивы. Число элементов и тип этих массивов определяются в соответствии с вариантами задания. Произвольные значения элементов массивов могут быть заданы любым доступным способом. Ширина поля каждой колонки должна подразумевать возможность вывода любого допустимого значения каждого элемента, причем для числовых данных необходимо учесть возможность вывода с максимальной точностью для физического представления текущего типа данных.

### Варианты задания

**1.** Даны: целочисленный массив из 8 коротких элементов; целочисленный массив из 10 коротких элементов; целочисленный массив из 12 элементов; вещественный массив двойной точности из 8 элементов; логический массив из 14 элементов; текстовый массив из 6 элементов по 11 символов.

**2.** Даны: целочисленный массив из 7 элементов; целочисленный массив из 5 длинных элементов; комплексный массив из 9 элементов; логический массив из 11 элементов; логический массив из 13 элементов; текстовый массив из 15 элементов по 5 символов.

**3.** Даны: целочисленный массив из 12 длинных элементов; вещественный массив из 15 элементов; вещественный массив двойной точности из 9 элементов; логический массив из 6 элементов; текстовый массив из 8 элементов по 2 символа; текстовый массив из 11 элементов по 2 символа.

4. Даны: целочисленный массив из 9 коротких элементов; целочисленный массив из 11 коротких элементов; целочисленный массив из 12 длинных элементов; комплексный массив из 15 элементов; логический массив из 7 элементов; текстовый массив из 10 элементов по 4 символа.

5. Даны: целочисленный массив из 9 коротких элементов; вещественный массив из 13 элементов; комплексный массив из 14 элементов; логический массив из 8 элементов; текстовый массив из 10 элементов по 2 символа; текстовый массив из 12 элементов по 2 символа.

6. Даны: целочисленный массив из 13 элементов; вещественный массив из 10 элементов; вещественный массив двойной точности из 7 элементов; логический массив из 5 элементов; текстовый массив из 15 элементов по 3 символа; текстовый массив из 17 элементов по 3 символа.

7. Даны: целочисленный массив из 8 длинных элементов; вещественный массив из 13 элементов; комплексный массив из 9 элементов; логический массив из 10 элементов; логический массив из 12 элементов; текстовый массив из 7 элементов по 5 символов.

8. Даны: целочисленный массив из 12 элементов; вещественный массив из 9 элементов; комплексный массив из 14 элементов; логический массив из 11 элементов; текстовый массив из 4 элементов по 2 символа; текстовый массив из 7 элементов по 2 символа.

9. Даны: целочисленный массив из 3 коротких элементов; целочисленный массив из 6 коротких элементов; целочисленный массив из 9 длинных элементов; вещественный массив из 12 элементов; логический массив из 8 элементов; текстовый массив из 6 элементов по 8 символов.

**10.** Даны: целочисленный массив из 13 коротких элементов; целочисленный массив из 11 длинных элементов; вещественный массив двойной точности из 6 элементов; логический массив из 8 элементов; текстовый массив из 10 элементов по 3 символа; текстовый массив из 12 элементов по 3 символа.

**11.** Даны: целочисленный массив из 16 коротких элементов; целочисленный массив из 14 элементов; вещественный массив двойной точности из 8 элементов; логический массив из 10 элементов; логический массив из 12 элементов; текстовый массив из 10 элементов по 10 символов.

**12.** Даны: целочисленный массив из 15 элементов; целочисленный массив из 11 длинных элементов; комплексный массив из 9 элементов; логический массив из 13 элементов; текстовый массив из 5 элементов по 5 символов; текстовый массив из 7 элементов по 5 символов.

**13.** Даны: целочисленный массив из 10 длинных элементов; вещественный массив из 13 элементов; вещественный массив двойной точности из 18 элементов; логический массив из 5 элементов; логический массив из 9 элементов; текстовый массив из 7 элементов по 7 символов.

**14.** Даны: целочисленный массив из 14 коротких элементов; целочисленный массив из 17 длинных элементов; комплексный массив из 8 элементов; логический массив из 11 элементов; логический массив из 13 элементов; текстовый массив из 9 элементов по 7 символов.

**15.** Даны: целочисленный массив из 12 коротких элементов; целочисленный массив из 14 коротких элементов; вещественный массив из 16 элементов; комплексный массив из 11 элементов; логический массив из 6 элементов; текстовый массив из 9 элементов по 3 символа.

**16.** Даны: целочисленный массив из 9 элементов; вещественный массив из 13 элементов; вещественный массив двойной точности из 11 элементов; логический массив из 5 элементов; логический массив из 7 элементов; текстовый массив из 11 элементов по 5 символов.

**17.** Даны: целочисленный массив из 7 длинных элементов; вещественный массив из 6 элементов; комплексный массив из 14 элементов; логический массив из 5 элементов; текстовый массив из 9 элементов по 3 символа; текстовый массив из 12 элементов по 3 символа.

**18.** Даны: целочисленный массив из 6 элементов; вещественный массив из 10 элементов; комплексный массив из 11 элементов; логический массив из 13 элементов; логический массив из 16 элементов; текстовый массив из 8 элементов по 5 символов.

**19.** Даны: целочисленный массив из 13 коротких элементов; целочисленный массив из 16 длинных элементов; вещественный массив из 9 элементов; логический массив из 5 элементов; логический массив из 7 элементов; текстовый массив из 13 элементов по 9 символов.

**20.** Даны: целочисленный массив из 12 коротких элементов; целочисленный массив из 14 коротких элементов; целочисленный массив из 15 длинных элементов; вещественный массив двойной точности из 11 элементов; логический массив из 7 элементов; текстовый массив из 8 элементов по 8 символов.

**21.** Даны: целочисленный массив из 19 коротких элементов; целочисленный массив из 15 элементов; вещественный массив двойной точности из 9 элементов; логический массив из 7 элементов; текстовый массив из 11 элементов по 7 символов; текстовый массив из 13 элементов по 7 символов.

**22.** Даны: целочисленный массив из 11 элементов; целочисленный массив из 18 длинных элементов; комплексный массив из 9 элементов; логический массив из 4 элементов; логический массив из 7 элементов; текстовый массив из 10 элементов по 9 символов.

**23.** Даны: целочисленный массив из 10 длинных элементов; вещественный массив из 16 элементов; вещественный массив двойной точности из 8 элементов; логический массив из 17 элементов; текстовый массив из 10 элементов по 3 символа; текстовый массив из 12 элементов по 3 символа.

**24.** Даны: целочисленный массив из 17 коротких элементов; целочисленный массив из 4 длинных элементов; комплексный массив из 11 элементов; логический массив из 10 элементов; текстовый массив из 12 элементов по 4 символа; текстовый массив из 14 элементов по 4 символа.

**25.** Даны: целочисленный массив из 10 коротких элементов; вещественный массив из 18 элементов; комплексный массив из 7 элементов; логический массив из 11 элементов; логический массив из 13 элементов; текстовый массив из 15 элементов по 8 символов.

**26.** Даны: целочисленный массив из 9 элементов; вещественный массив из 18 элементов; вещественный массив двойной точности из 7 элементов; логический массив из 14 элементов; текстовый массив из 11 элементов по 2 символа; текстовый массив из 13 элементов по 2 символа.

**27.** Даны: целочисленный массив из 13 длинных элементов; вещественный массив из 8 элементов; комплексный массив из 4 элементов; логический массив из 16 элементов; логический массив из 18 элементов; текстовый массив из 10 элементов по 3 символа.

**28.** Даны: целочисленный массив из 11 элементов; вещественный массив из 16 элементов; комплексный массив из 6 элементов; логический массив из 13 элементов; текстовый массив из 17 элементов по 4 символа; текстовый массив из 19 элементов по 4 символа.

**29.** Даны: целочисленный массив из 12 коротких элементов; целочисленный массив из 14 коротких элементов; целочисленный массив из 16 длинных элементов; вещественный массив из 11 элементов; логический массив из 8 элементов; текстовый массив из 8 элементов по 10 символов.

**30.** Даны: целочисленный массив из 19 коротких элементов; целочисленный массив из 17 длинных элементов; вещественный массив двойной точности из 7 элементов; логический массив из 9 элементов; логический массив из 11 элементов; текстовый массив из 12 элементов по 6 символов.

**Пример 8.** Даны:

- a)** логический массив из 4 элементов;
- b)** целочисленный массив из 3 коротких элементов;
- c)** целочисленный массив из 4 элементов;
- d)** вещественный массив одинарной степени точности из 5 элементов;
- e)** целочисленный массив из 4 длинных элементов;
- f)** вещественный массив двойной степени точности из 5 элементов;
- g)** текстовый массив из 3 элементов по 4 символа.

Требуется составить программу для ЭВМ, позволяющую сформировать две таблицы, колонки которых содержат массивы **a) - d)** и **e) - g)** соответственно.

Любые элементы массива **a)** могут быть отображены в 1 символ, массива **b)** - в 4 символа, массива **c)** - в 6 симво-

лов, массива **d**) - в экспоненциальной форме с 8 значащими цифрами, массива **e**) - в 11 символов, массива **f**) - в экспоненциальной форме с 16 значащими цифрами, массива **g**) - в 4 символа.

С ЗАДАНИЕ 8. Пример 8. Прогр. на языке FORTRAN.

```

PROGRAM T8
*
PARAMETER ( L = 3, M = 4, N = 5 )
LOGICAL*1 LL(M)
INTEGER*1 IS(L)
INTEGER*2 IM(M)
REAL*4 RR(N)
INTEGER*4 IL(M)
REAL*8 DD(N)
CHARACTER*4 TT(L)
*
DATA LL / 2*.FALSE., 2*.TRUE. /
DATA IS / -128, 0, 127 /
DATA IM / -32768, 0, 1, 32767 /
DATA RR / -3.4028234E38, -333333.33,
*          1.5E-45, 1.E-38, 3.4E38 /
*
DATA IL / -2147483647, 0, 1, 2147483647 /
DATA DD / -1.7D308, -1.111111111111111D0,
*          -5.D-324, 1.D-308, 1.79769D308/
DATA TT / 'A_B', 'A_B_', '_AB_' /
*
WRITE(*,*)
WRITE(*,1)
WRITE(*,11)

```



```

WRITE(*,1)
1  FORMAT(5X,42('-'))
11  FORMAT(5X,'| N | LL| IS | IM |',
*      7X,'RR',7X,'|')
DO I = 1, L
WRITE(*,22) I, LL(I), IS(I), IM(I), RR(I)
END DO
22  FORMAT(5X,'|',I2,' |',L2,' |',I5,' |',I7,
*      ' |',E15.8,' |')
*
DO I = L+1, M
WRITE(*,33) I, LL(I), IM(I), RR(I)
END DO
33  FORMAT(5X,'|',I2,' |',L2,' |',I7,
*      ' |',E15.8,' |')
*
DO I = M+1, N
WRITE(*,44) I, RR(I)
END DO
44  FORMAT(5X,'|',I2,' |',I7,7X,
*      ' |',E15.8,' |')
WRITE(*,1)
*
WRITE(*,*)
WRITE(*,55)
WRITE(*,66)
WRITE(*,55)
55  FORMAT(1X,51('-'))
66  FORMAT(' | N | ',5X,'IL',5X,'|',11X,'DD',11X,
*      '| TT |')
*

```

```

DO I = 1, L
  WRITE(*,77) I, IL(I), DD(I), TT(I)
END DO
77  FORMAT(' |',I2,' |',I12,' |',D23.16,' |',
*        A5,' |')
*
DO I = L+1, M
  WRITE(*,88) I, IL(I), DD(I)
END DO
88  FORMAT(' |',I2,' |',I12,' |',D23.16,' |',
*        5X,' |')
*
DO I = M+1, N
  WRITE(*,99) I, DD(I)
END DO
99  FORMAT(' |',I2,' |',12X,' |',D23.16,' |',
*        5X,' |')
*
WRITE(*,55)
STOP
END

```

```

-----
| N | LL|  IS |  IM |      RR      |
-----
| 1 | F | -128 | -32768 | -.34028230E+39 |
| 2 | F |   0  |    0   | -.33333330E+06 |
| 3 | T |  127 |    1   | .14012980E-44 |
| 4 | T |     | 32767 | .99999990E-38 |
| 5 |   |     |     | .34000000E+39 |
-----

```

N	IL	DD	TT
1	-2147483647	-.17000000000000000+309	A__B
2	0	-.11111111111111111D+01	A_B_
3	1	-.4940656458412465-323	_AB_
4	2147483647	.9999999999999999-308	
5		.1797690000000000+309	

/\* ЗАДАНИЕ 8. Пример 8. Программа на языке C. \*/

```
#include <stdio.h>
#include <math.h>

int main( void ) {

#define l 3
#define m 4
#define n 5

int i;

char ll[m] = { 'f', 'f', 't', 't' };
short is[l] = { -128, 0, 127 };
int im[m] = { -32768, 0, 1, 32767 };
float rr[n] = { -3.4028234e38, -333333.33,
               1.4e-45, 1.1754944e-38, 3.4e38 };

long il[m] = { -21474836481, 0, 1, 21474836471 };
```

```

double dd[n] = { -1.797693e308, -1.1111111111111111,
                 5.e-324, 2.22507386e-308, 1.7e308 };
char   tt[l][m] = { "A_B", "A_B_", "_AB_" };

printf( "\n" ); printf(
    "-----\n");
printf(
    "| n | ll| is | im | rr | \n");
printf(
    "-----\n");

for( i = 0; i < l; i++ ) {
    printf( "%2d |%2c |%5d |%7d |% 15.7e | \n",
           i+1, ll[i], is[i], im[i], rr[i] );
}
for( i = 1; i < m; i++ ) {
    printf( "%2d |%2c | %7d |% 15.7e | \n",
           i+1, ll[i], im[i], rr[i] );
}
for( i = m; i < n; i++ ) {
    printf( "%2d | | | % 15.7e | \n",
           i+1, rr[i] );
} printf(
    "-----\n");

printf( "\n" );
for( i = 1; i < 52; i++ ) printf( "-");
printf( "\n" );

printf( "| n | il | dd" );
printf( " | tt | \n");

```

```

for( i = 1; i < 52; i++ ) printf( "-");
printf( "\n" );

for( i = 0; i < l; i++ ) {
printf( "%2d %12ld %23.15e %5.4s |\n",
i+1, il[i], dd[i], tt[i] );
}
for( i = 1; i < m; i++ ) {
printf( "%2d %12ld %23.15e |          |\n",
i+1, il[i], dd[i] );
}
for( i = m; i < n; i++ ) {
printf( "%2d |          |%23.15e |          |\n",
i+1, dd[i] );
}
for( i = 1; i < 52; i++ ) printf( "-");
printf( "\n" );

return 1;
}

```

```

-----
| n | ll|  is |   im |         rr          |
-----
| 1 | f | -128 | -32768 | -3.4028235e+38 |
| 2 | f |   0 |     0 | -3.3333334e+05 |
| 3 | t |  127 |     1 | 0.0000000e+00 |
| 4 | t |     |  32767 | 1.1754944e-38 |
| 5 |   |     |     | 3.4000000e+38 |
-----

```

n	il	dd	tt
1	-2147483648	-1.7976930000000000e+308	A__B
2	0	-1.1111111111111111e+00	A_B_
3	1	0.0000000000000000e+00	_AB_
4	2147483647	2.2250738600000000e-308	
5		1.7000000000000000e+308	

## ЗАДАНИЕ 9

Представить рациональные числа вида  $x = a/b$  (конечные либо периодические десятичные дроби) в позиционных системах счисления с основанием  $r$ . Значения чисел и оснований позиционных систем счисления определяются в соответствии с вариантами задания.

### Варианты задания

1.  $x = 17/3$ ;  $r = 2, 3, 10, 16$ .
2.  $x = 19/5$ ;  $r = 2, 5, 10, 14$ .
3.  $x = 23/7$ ;  $r = 2, 7, 10, 12$ .
4.  $x = 29/11$ ;  $r = 2, 10, 11, 15$ .
5.  $x = 31/13$ ;  $r = 2, 10, 11, 13$ .
6.  $x = 37/3$ ;  $r = 2, 3, 10, 16$ .
7.  $x = 41/5$ ;  $r = 2, 5, 10, 14$ .
8.  $x = 43/7$ ;  $r = 2, 7, 10, 12$ .
9.  $x = 47/11$ ;  $r = 2, 10, 11, 15$ .
10.  $x = 53/13$ ;  $r = 2, 10, 11, 13$ .
11.  $x = 59/3$ ;  $r = 2, 3, 10, 16$ .
12.  $x = 61/5$ ;  $r = 2, 5, 10, 14$ .
13.  $x = 67/7$ ;  $r = 2, 7, 10, 12$ .
14.  $x = 71/11$ ;  $r = 2, 10, 11, 15$ .
15.  $x = 73/13$ ;  $r = 2, 10, 11, 13$ .
16.  $x = 79/3$ ;  $r = 2, 3, 10, 16$ .
17.  $x = 83/5$ ;  $r = 2, 5, 10, 14$ .
18.  $x = 89/7$ ;  $r = 2, 7, 10, 12$ .
19.  $x = 91/11$ ;  $r = 2, 10, 11, 15$ .
20.  $x = 97/13$ ;  $r = 2, 10, 11, 13$ .
21.  $x = 17/3$ ;  $r = 3, 4, 10, 16$ .

- |     |              |                      |
|-----|--------------|----------------------|
| 22. | $x = 19/5;$  | $r = 5, 6, 10, 14.$  |
| 23. | $x = 23/7;$  | $r = 7, 8, 10, 12.$  |
| 24. | $x = 29/11;$ | $r = 3, 10, 11, 15.$ |
| 25. | $x = 31/13;$ | $r = 4, 10, 11, 13.$ |
| 26. | $x = 37/3;$  | $r = 3, 4, 10, 16.$  |
| 27. | $x = 41/5;$  | $r = 5, 6, 10, 14.$  |
| 28. | $x = 43/7;$  | $r = 7, 8, 10, 12.$  |
| 29. | $x = 47/11;$ | $r = 3, 10, 11, 15.$ |
| 30. | $x = 53/13;$ | $r = 4, 10, 11, 13.$ |
| 31. | $x = 59/3;$  | $r = 3, 4, 10, 16.$  |
| 32. | $x = 61/5;$  | $r = 5, 6, 10, 14.$  |
| 33. | $x = 67/7;$  | $r = 7, 8, 10, 12.$  |
| 34. | $x = 71/11;$ | $r = 3, 10, 11, 15.$ |
| 35. | $x = 73/13;$ | $r = 4, 10, 11, 13.$ |
| 36. | $x = 79/3;$  | $r = 3, 4, 10, 16.$  |
| 37. | $x = 83/5;$  | $r = 5, 6, 10, 14.$  |
| 38. | $x = 89/7;$  | $r = 7, 8, 10, 12.$  |
| 39. | $x = 91/11;$ | $r = 3, 10, 11, 15.$ |
| 40. | $x = 97/13;$ | $r = 4, 10, 11, 13.$ |

**Пример 9.** Представить рациональное число  $x = 97/7$  в позиционных системах счисления с основаниями  $r = 2 - 16$ .

$$97/7 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + \dots =$$

$$= 1101.110\dots_2 =$$

$$= 1 \cdot 3^2 + 1 \cdot 3^1 + 1 \cdot 3^0 + 2 \cdot 3^{-1} + 1 \cdot 3^{-2} + 2 \cdot 3^{-3} + 0 \cdot 3^{-4} +$$

$$+ 1 \cdot 3^{-5} + 0 \cdot 3^{-6} + \dots = 111.212010\dots_3 =$$



$$= 3 \cdot 4^1 + 1 \cdot 4^0 + 3 \cdot 4^{-1} + 1 \cdot 4^{-2} + 2 \cdot 4^{-3} + \dots = 31.312\dots_4 =$$

$$= 2 \cdot 5^1 + 3 \cdot 5^0 + 4 \cdot 5^{-1} + 1 \cdot 5^{-2} + 2 \cdot 5^{-3} + 0 \cdot 5^{-4} + \\ + 3 \cdot 5^{-5} + 2 \cdot 5^{-6} + \dots = 23.412032\dots_5 =$$

$$= 2 \cdot 6^1 + 1 \cdot 6^0 + 5 \cdot 6^{-1} + 0 \cdot 6^{-2} + \dots = 21.50\dots_6 =$$

$$= 1 \cdot 7^1 + 6 \cdot 7^0 + 6 \cdot 7^{-1} = 16.6_7 =$$

$$= 1 \cdot 8^1 + 5 \cdot 8^0 + 6 \cdot 8^{-1} + \dots = 15.6\dots_8 =$$

$$= 1 \cdot 9^1 + 4 \cdot 9^0 + 7 \cdot 9^{-1} + 6 \cdot 9^{-2} + 3 \cdot 9^{-3} + \dots = 14.763\dots_9 =$$

$$= 1 \cdot 10^1 + 3 \cdot 10^0 + 8 \cdot 10^{-1} + 5 \cdot 10^{-2} + 7 \cdot 10^{-3} + 1 \cdot 10^{-4} + \\ + 4 \cdot 10^{-5} + 2 \cdot 10^{-6} + \dots = 13.857142\dots_{10} =$$

$$= 1 \cdot 11^1 + 2 \cdot 11^0 + 9 \cdot 11^{-1} + 4 \cdot 11^{-2} + 7 \cdot 11^{-3} + \dots = 12.947\dots_{11} =$$

$$= 1 \cdot 12^1 + 1 \cdot 12^0 + 10 \cdot 12^{-1} + 3 \cdot 12^{-2} + 5 \cdot 12^{-3} + 1 \cdot 12^{-4} +$$

$$\begin{aligned}
& +8 \cdot 12^{-5} + 6 \cdot 12^{-6} + \dots = 11.A35186\dots_{12} = \\
& = 1 \cdot 13^1 + 0 \cdot 13^0 + 11 \cdot 13^{-1} + 1 \cdot 13^{-2} + \dots = 10.B1\dots_{13} = \\
& \quad = 13 \cdot 14^0 + 12 \cdot 14^{-1} = D.C_{14} = \\
& \quad = 13 \cdot 15^0 + 12 \cdot 15^{-1} + \dots = D.C\dots_{15} = \\
& = 13 \cdot 16^0 + 13 \cdot 16^{-1} + 11 \cdot 16^{-2} + 6 \cdot 16^{-3} + \dots = D.DB6\dots_{16} .
\end{aligned}$$

## ЛИТЕРАТУРА

1. **Грунд Ф.** Программирование на языке ФОРТРАН IV / Пер. с нем. - М.: Мир, 1976. - 184 с.
2. **Калдербенк В. Д.** Курс программирования на ФОРТРАНе - IV / Пер. с англ. - М.: Энергия, 1978. - 87 с.
3. **Катцан Г.** Язык Фортран 77 / Пер. с англ. - М.: Мир, 1982. - 208 с.
4. **Браух В.** Программирование на Фортране 77 для инженеров / Пер. с нем. - М.: Мир, 1987. - 200 с.
5. **Джехани Н.** Программирование на языке Си / Пер. с англ. - М.: Радио и связь, 1988. - 270 с.
6. **Болски М.И.** Язык программирования Си: Справочник / Пер. с англ. - М.: Радио и связь, 1988. - 96 с.
7. **Керниган Б., Ритчи Д.** Язык программирования Си / Пер. с англ. - М.: Финансы и статистика, 1992. - 272 с.

