Задание. Преобразовать число 1110100,10(B) в десятичное, восьмеричное и шестнадцатеричное.

1110100,10(B)=$1*2^6+1*2^5+1*2^4+0*2^3+1*2^2+0*2^1+0*2^0+1*2^{-1}+0*2^{-2}$=
=1*64+1*32+1*16+0*8+1*4+0*2+0*1+1*(1/2)+0*(1/4)=64+32+16+4+1/2=116,5(D)

001(B)=1(O), 110(B)=6(O), 100(B)=4(O) -> 001 110 100,100(B)=164,4(O)

0111(B)=7(H), 0100(B)=4(H), 1000(B)=8(H) -> 0111 0100,1000(B)=74,8(H)

Задание. Перевести числа из одной системы счисления в другую.

B ->D: 10101111

10101111(B)=$1*2^0+1*2^1+1*2^2+1*2^3+0*2^4+1*2^5+0*2^6+1*2^7$=1*1+1*2+1*4+1*8+0*16+1*32 +0*64+1*128=1+2+4+8+32+128=175(D)

D ->B: 102

```
1 0 2 |2
1 0 2 |5 1 |2
    0  5 0 |2 5 |2
        1  2 4 |1 2 |  2
            1  1 2 |    6 |2
                0       6 |  3 |2
                        0    2 |   1
                             1
```

102(D)=1100110(B)

H ->D: 5E

5E(H)=$14*16^0+5*16^1$=14*1+5*16=14+80=94(D)

D ->H: 401

```
4 0 1 |1 6
4 0 0 |2 5 |1 6
    1  1 6 |   1
         9
```

401(D)=191(H)

B ->H: 10101011

1010(B)=A(H); 1011(B)=B(H)

10101011(B)=AB(H)

H ->B: 5AC

5(H)=101(B); A(H)=1010(B); C(H)=1100(B)

5AC(H)=10110101100(B)

Задание. Перевести числа X1=77 и X2=-65 из десятичной системы в двоичную, сложить два числа в дополнительном коде, перевести результат в прямой код, обратно в десятичную систему счисления, проверить. Формат числа : 1 разряд знак, 7 разрядов значащих.

77 | 2
76 | 38 | 2
1 38 | 19 | 2
0 18 | 9 | 2
1 8 | 4 | 2
1 4 | 2 | 2
0 2 | **1**
0

X1=01001101(прямой код)

65 | 2
64 | 32 | 2
1 32 | 16 | 2
0 16 | 8 | 2
0 8 | 4 | 2
0 4 | 2 | 2
0 2 | **1**
0

X2=11000001(прямой код)

X2=10111111(дополнительный код)

$$\begin{array}{r} 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1 \\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1 \\ \hline 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0 \end{array}$$

X1+X2=00001100(В прямой код) $=0*2^0+0*2^1+1*2^2+1*2^3=4+8=12$(D).

Проверка X1+X2=77+(-65)=12.

Задание. Записать числа X3=-32.25(D) и X4=-475.0 (D) в формате с плавающей запятой (32 разряда) для процессора FPU Intel.

$X3=-32.25(D)=-100000.01(B)= (-1)^{1}*1.0000001\ (B)*2^{5(D)}$
EXP=e+127(D)=5(D)+127(D)=132(D)=10000100(B)
S=1
$X3=\underbrace{1}_{S}\underbrace{10000100}_{EXP}\underbrace{0000000100000000000000000000000000000000000000}_{FRACT}(B)=C2010000(H)$

$X4=475.0\ (D)=111011011(B)= (-1)^{1}*1.11011011\ (B)*2^{8(D)}$
EXP=e+127(D)=8(D)+127(D)=135(D)=10000111(B)
S=1
$X4=\underbrace{0}_{S}\underbrace{10000111}_{EXP}\underbrace{110110110000000000000000000000000000000000000}_{FRACT}(B)=C3ED8000(H)$

Задание. Перевести числа X5=C2BB8000(H) и X6=42C68000(H) из формата FPU32 в десятичную систему счисления.

$X5= C2BB8000(H)=\underbrace{1}_{S}\underbrace{10000101}_{EXP}\underbrace{01110111000000000000000000000000000000000000}_{FRACT}(B)$

EXP=10000101(B)=133(D)

e=EXP-127(D)=133(D)-127(D)=6(D)

$X5=(-1)^{S}*1.FRACT*2^{e}=(-1)^{1}*1.\ 01110111(B)*2^{6(D)}= -1011101.11(B)=-95.75(D)$

$X6=\ 42C68000(H)=\underbrace{0}_{S}\underbrace{10000101}_{EXP}\underbrace{1000110100000000000000000000000000000000000}_{FRACT}(B)$

EXP=10000101(B)=133(D)

e=EXP-127(D)=133(D)-127(D)=6(D)

$X5=(-1)^{S}*1.FRACT*2^{e}=(-1)^{0}*1.\ 10001101(B)*2^{6(D)}=1100011.01(B)=99.25(D)$

Задание. Округлите сомнительные цифры числа *a*, оставив верные цифры: а) в узком смысле; б) в широком смысле. Определите абсолютную погрешность результата, если

а) a=62,8482 (±0,0044); б) a=4,177, δ_a=0,16%.

Решение

а) По условию $\Delta_a = 0{,}0044 < 0{,}005 = 0{,}5 \cdot 10^{-2}$, следовательно, в числе a=62,8482 верными в узком смысле являются четыре цифры: 6, 2, 8, 4. Округляем число a до четырех значащих цифр: $a \approx a_1 = 62{,}85$. Тогда $\Delta_{a_1} = \Delta_a + \Delta_{a,a_1} = 0{,}0044 + 0{,}0018 = 0{,}0062$.

Так как $\Delta_{a_1} = 0{,}0062 < 0{,}05 = 0{,}5 \cdot 10^{-1}$, то число a_1 имеет три верные цифры: 6, 2, 8. Округляем число a до трех значащих цифр: $a \approx a_2 = 62{,}9$. Тогда $\Delta_{a_2} = \Delta_a + \Delta_{a,a_2} = 0{,}0044 + 0{,}0518 = 0{,}0562$.

Так как $\Delta_{a_2} = 0{,}0562 < 0{,}5 = 0{,}5 \cdot 10^{0}$, то число a_2 имеет две верные цифры: 6, 2. Округляем число a до двух значащих цифр: $a \approx a_3 = 63$. Тогда $\Delta_{a_3} = \Delta_a + \Delta_{a,a_3} = 0{,}0044 + 0{,}1518 = 0{,}1562$.

Так как $\Delta_{a_2} = 0{,}1562 < 0{,}5 = 0{,}5 \cdot 10^{0}$, то две оставшиеся цифры результата $a_3 = 63$ верны в узком смысле. Таким образом, $a \approx 63(\pm 0{,}1562)$.

б) Представим δ_a в виде δ_a=0,0016 и найдем $\Delta_a = |a| \cdot \delta_a = 4{,}177 \cdot 0{,}0016 = 0{,}0066832 < 0{,}0067$; примем $\Delta_a = 0{,}0067$. Так как $\Delta_a = 0{,}0067 < 0{,}01 = 10^{-2}$, что число a=4,177 имеет три верные в широком смысле цифры: 4, 1, 7. Округляем число a до трех значащих цифр: $a \approx a_1 = 4{,}18$. Тогда $\Delta_{a_1} = \Delta_a + \Delta_{a,a_1} = 0{,}0067 + 0{,}003 = 0{,}0097$.

Так как $\Delta_{a_1} = 0{,}0097 < 0{,}01 = 10^{-2}$, то три оставшиеся цифры результата $a_1 = 4{,}18$ верны в широком смысле. Таким образом, $a \approx 4{,}18(\pm 0{,}0097)$.

Ответ:

а) $a \approx a_3 = 63, \quad \Delta_{a_3} = 0{,}1562$;

б) $a \approx a_1 = 4{,}18, \quad \Delta_{a_1} = 0{,}0097$.

Задание. Найдите интерполяционный многочлен Лагранжа для функции $f(x)=\frac{1}{x+1}$ с узлами x_0=-3, x_1=-2, x_2=0, x_3=1.

Решение

Прежде всего, заметим, что y_0=-1/2, y_1=-1, y_2=1, y_3=1/2.

Применяя формулу для вычисления многочлена Лагранжа при n=3, получим:

$$L_3(x)=y_0\frac{(x-x_1)\cdot(x-x_2)\cdot(x-x_3)}{(x_0-x_1)\cdot(x_0-x_2)\cdot(x_0-x_3)}+y_1\frac{(x-x_0)\cdot(x-x_2)\cdot(x-x_3)}{(x_1-x_0)\cdot(x_1-x_2)\cdot(x_1-x_3)}+$$

$$+y_2\frac{(x-x_0)\cdot(x-x_1)\cdot(x-x_3)}{(x_2-x_0)\cdot(x_2-x_1)\cdot(x_2-x_3)}+y_3\frac{(x-x_0)\cdot(x-x_1)\cdot(x-x_2)}{(x_3-x_0)\cdot(x_3-x_1)\cdot(x_3-x_2)}=$$

$$=-\frac{1}{2}\cdot\frac{(x-(-2))\cdot(x-0)\cdot(x-1)}{(-3-(-2))\cdot(-3-0)\cdot(-3-1)}-\frac{(x-(-3))\cdot(x-0)\cdot(x-1)}{(-2-(-3))\cdot(-2-0)\cdot(-2-1)}+$$

$$+\frac{(x-(-3))\cdot(x-(-2))\cdot(x-1)}{(0-(-3))\cdot(0-(-2))\cdot(0-1)}+\frac{1}{2}\cdot\frac{(x-(-3))\cdot(x-(-2))\cdot(x-0)}{(1-(-3))\cdot(1-(-2))\cdot(1-0)}=$$

$$=\frac{1}{24}\cdot(x+2)\cdot x\cdot(x-1)-\frac{1}{6}\cdot(x+3)\cdot x\cdot(x-1)-$$

$$-\frac{1}{6}\cdot(x+3)\cdot(x+2)\cdot(x-1)+\frac{1}{24}\cdot(x+3)\cdot(x+2)\cdot x=$$

$$=-\frac{1}{4}x^3-\frac{3}{4}x^2+\frac{1}{2}x+1$$

Ответ: $L_3(x)=-0{,}25x^3-0{,}75x^2+0{,}5x+1$.

Задание. Методом наименьших квадратов найдите эмпирическую формулу вида $y=ax+b$ по данным опыта, представленным таблицей

x	1	2	3	4	5
y	2,8	2,3	4,3	5,8	4,8

Решение

Результаты предварительных вычислений внесем в таблицу.

k	x_k	y_k	$x_k y_k$	x^2_k
0	1	2,8	2,8	1
1	2	2,3	4,6	4
2	3	4,3	12,9	9
3	4	5,8	23,2	16
4	5	4,8	24	25
$\sum$	15	20	67,5	55

Нормальная система уравнений в данном случае принимает вид

$$\begin{cases} 55a+15b=67{,}5 \\ 15a+5b=20 \end{cases}$$

Отсюда находим: *a=0,75, b=1,75*. Следовательно, искомая эмпирическая формула имеет вид $y=0{,}75x+1{,}75$ (рис.1)

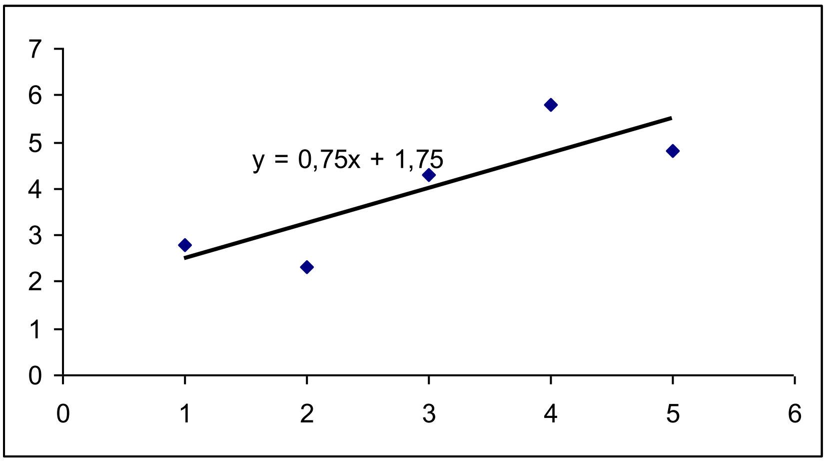


Рис.1

Ответ: $y=0{,}75x+1{,}75$

Задание. Вычислите интеграл $\int_0^1 \sqrt{2+x^3}\,dx$ с помощью формулы Симпсона, разбив отрезок интегрирования на 10 частей. Вычисления производите с округлением до четвертого десятичного знака.

Решение

Нужно определить значения подынтегральной функции при $h = \dfrac{b-a}{n} = 0.1$ для следующих значений аргумента: x_0=0; x_1=0.1; x_2=0.2; x_3=0.3; x_4=0.4; x_5=0.5; x_6=0.6; x_7=0.7; x_8=0.8; x_9=0.9; x_{10}=1.

Находим соответствующие значения функции $f(x) = \sqrt{2+x^3}$: $y_0 = \sqrt{2} = 1{,}4142$; $y_1 = \sqrt{2.001} \approx 1.4146$; $y_2 = \sqrt{2.008} \approx 1.4170$; $y_3 = \sqrt{2.027} \approx 1.4237$; $y_4 = \sqrt{2.064} \approx 1.4367$; $y_5 = \sqrt{2.125} \approx 1.4577$; $y_6 = \sqrt{2.216} \approx 1.4886$; $y_7 = \sqrt{2.343} \approx 1.5307$; $y_8 = \sqrt{2.512} \approx 1.5849$; $y_9 = \sqrt{2.729} \approx 1.6520$; $y_{10} = \sqrt{3} \approx 1.7321$.

Тогда по формуле Симпсона получаем:

$$\int_0^1 \sqrt{2+x^3}\,dx \approx \frac{0.1}{3}(1.4142 + 1.7321 + 4(1.4146 + 1.4237 + 1.4577 + 1.5307 + 1.6520) +$$
$$+ 2(1.4170 + 1.4367 + 1.4886 + 1.5849)) = 1.49718 \approx 1.4972$$

Ответ: $\int_0^1 \sqrt{2+x^3}\,dx \approx 1.4972$

Задание. Отделите корни уравнения $4x^3 + x^2 - 19x + 1 = 0$ аналитически и уточните больший из них методом Ньютона с точностью до $\Delta=10^{-3}$.

Решение

Отделим корни данного уравнения аналитически.

Находим:

$f(x) = 4x^3 + x^2 - 19x + 1$; $f'(x) = 12x^2 + 2x - 19$;

$f'(x) = 0$; $x_{1,2} = \dfrac{-2 \pm \sqrt{2^2 - 4*12*(-19)}}{2*12}$; x_1=-1.344..; x_2=1.1777…

Составим таблицу знаков функции f(x).

x	-∞	-1.3	1.2	+∞
sign f(x)	-	+	-	+

Итак, уравнение имеет три действительных корня:

$\xi_1 \in (-\infty;-1.3); \xi_2 \in (-1.3;1.2); \xi_3 \in (1.2;+\infty)$.

Уменьшим отрезки, содержащие корни, до длинны, равной 1.

x	-3	-2	-1	0	1	2	3
sign f(x)	-	+	+	+	-	-	+

Значит, $\xi_1 \in [-3;-2]; \xi_2 \in [0;1]; \xi_3 \in [2;3]$.

Уточним больший корень ξ_3 заданного уравнения методом Ньютона.

Имеем, $f'(x) = 12x^2 + 2x - 19 > 0$, $f''(x) = 24x + 2 > 0$, при $x \in [2,3]$, поэтому, чтобы воспользоваться методом Ньютона, следует выбрать x_0=3. Вычисления будем вести по формуле $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$, где f(3)=61, f'(3)=95.

Сведем вычисления в следующую таблицу.

k	x_k	$f(x_k)$	$f'(x_k)$	$f(x_k)/ f'(x_k)$	$x_{k+1}= x_k -f(x_k)/ f'(x_k)$
0	3,0000	61,0000	95,0000	0,6421	2,3579
1	2,3579	14,1961	52,4318	0,2708	2,0871
2	2,0871	2,0681	37,4482	0,0552	2,0319
3	2,0319	0,0788	34,6079	0,0023	2,0296
4	2,0296	0,0001	34,4925	0	2,0296

Из таблицы видно, что искомый корень $\xi_3 \approx 2.029$

Ответ: $\xi_1 \in [-3;-2]; \xi_2 \in [0;1]; \xi_3 \approx 2.029$

Задание. Используя метод Эйлера, составьте таблицу приближенных значений дифференциального уравнения $y' = \frac{2y}{x} + \frac{1}{x^3}$, удовлетворяющего начальному условию y(1)=0, на отрезке [1;1.05] с шагом h=0.01. Вычисления ведите с четырьмя знаками после запятой.

Решение

Находим последовательные значения аргумента: x_0=1; x_1=1.01; x_2=1.02; x_3=1.03; x_4=1.04; x_5=1.05.

Воспользуемся формулой $y_{k+1} = y_k + hf(x_k, y_k)$. Обозначим $f(x, y) = \frac{2y}{x} + \frac{1}{x^3}$

Для удобства вычислений составим таблицу.

k	x_k	y_k	$f(x_k,y_k)$	$y_{k+1}= y_k+h* f(x_k,y_k)$
0	1,00	0	1	0,01
1	1,01	0,0100	0,9904	0,0199
2	1,02	0,0199	0,9813	0,0297
3	1,03	0,0297	0,9728	0,0394
4	1,04	0,0394	0,9649	0,0491
5	1,05	0,0491	0,9574	0,0587

Ответ:

x_k	1,00	1,01	1,02	1,03	1,04	1,05
y_k	0	0,01	0,0199	0,0297	0,0394	0,0491

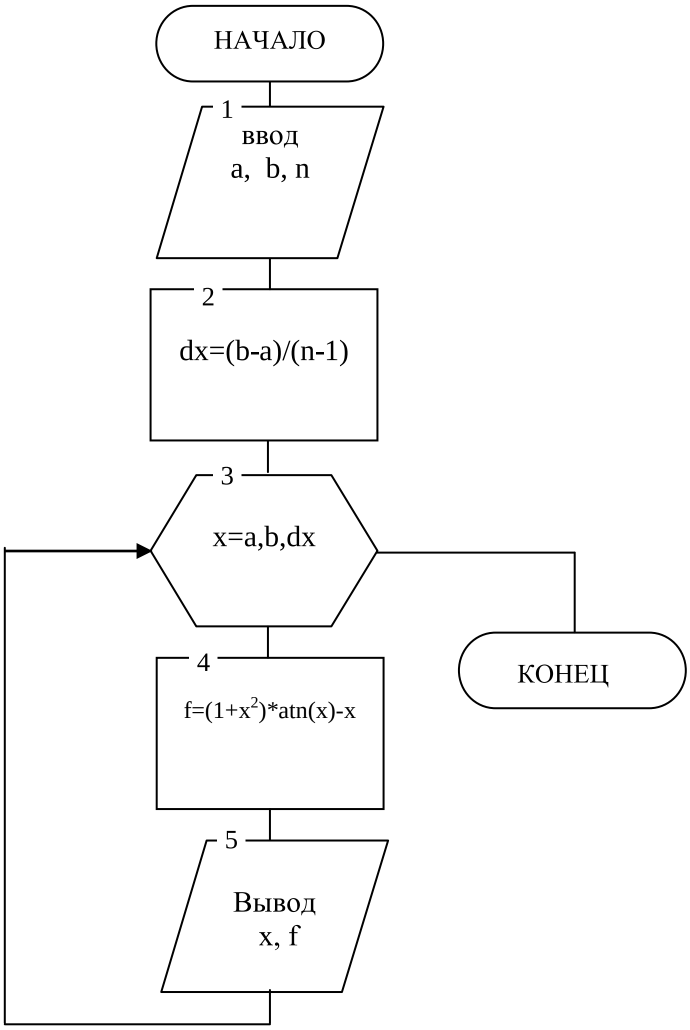
Задание. Составить программный код для вычисления значения функции:

$$P = \begin{cases} b * z, & если \quad z < -2 \\ z + b, & если \quad -2 \le z < 0 \\ z^2, & если \quad z \ge 0 \end{cases}.$$

```
Function P(b, z)
Rem проверка условия z < (-2)
If z < (-2) Then
Rem при его выполнении вычисляем значение P
   P = b * z
Else
            Rem проверка условия z >=2
   If z >= 2 Then
   Rem при его выполнении вычисляем значение P
      P = z ^ 2
   Else
   Rem при невыполнении предыдущих условий вычисляем значение P
   P = z + b
   End If
End If
End Function
```

Задание. Вычислить функцию F(x) на отрезке [a,b] с шагом h=(b-a)/(n-1); $F(x)=(1+x^2)arctg(x)-x$; a=-1; b=2; n=26.

Структурная схема алгоритма



Листинг программы на языке BASIC

```
open "zadan_1.txt" for output as #1
print #1, "Задание 1, вариант 20"
print #1, "Результат расчета значений функции на заданном отрезке"
print "Введите начальную и конечную точки отрезка"
input A,B
print #1, "Задан отрезок для расчета от ";A;" до ";B
print "Введите число разбиений отрезка "
input N
print #1, "Отрезок разбит на ";N;" частей"
rem вычисление шага изменения значения X
DX=(B-A)/(N-1)
print #1, "Функция f(x)=(1+x^2)*atn(x)-x"
def FNF(X)=(1+X^2)*atn(X)-X
for X=A to B step DX
print #1,using "x=##.## f(x)=##.##";X;FNF(X)
next X
close #1
end
```

Задание. Построить схему машинного алгоритма и составить Паскаль-программу решения следующей задачи с помощью оператора выбора Case.

$$\varphi_6 = \begin{cases} t^2 + tg\,\omega^2 + 3 & при \quad \omega < 0{,}6; \\ \sin\sqrt{t} + \ln^2 x_1 & при \quad \omega = 0{,}6; \\ e^{\beta t} - \cos\sqrt{t+\omega} & при \quad \omega > 0{,}6. \end{cases}$$

Для каждой из ветвей значение селектора вычислять по формуле N=2*K+6, K=1..3.

Исходные данные указаны приведенными ниже значениями:

$t = 0{,}8 \cdot 10^2$; ω=0,37; $x_1 = 1{,}752; \beta = -1{,}1.$

Алгоритм решения

Разрабатываемая программа должна запросить у пользователя исходные данные. В зависимости от значения параметра ω программа должна вычислить значение искомой функции. Номер ветви, по которой программа выполняла вычисление (К), используется для расчета селектора (N).

Результаты вычисления искомой функции и селектора (N) должны быть выведены на экран.

Схема машинного алгоритма

Приведенная ниже схема машинного алгоритма отражает вычислительную структуру решаемой задачи:

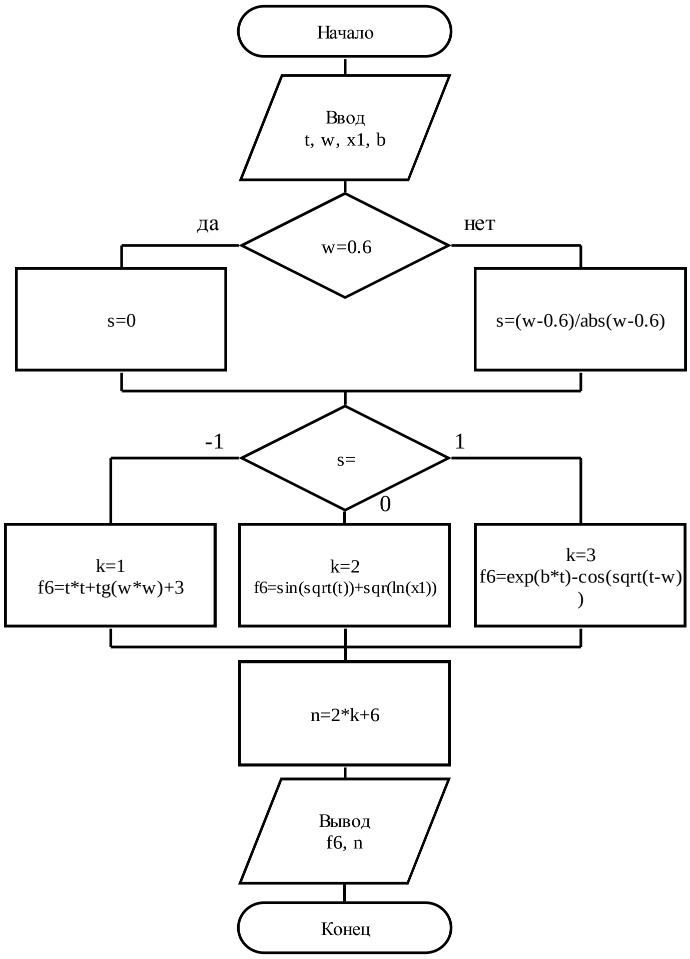


Таблица идентификаторов

Данная таблица устанавливает связь между исходными параметрами задачи и соответствующими им обозначениями в Паскаль-программе:

ω	t	x_1	φ_6	N	β	K
w	t	x1	f6	n	b	k

Паскаль-программа

```
Program  lr;
(*подключение модуля управления экраном*)
Uses wincrt;
(*раздел описания переменных*)
Var
    t,w,x1,b,f6:extended;
    s,k,n : integer;
(*раздел операторов*)
Begin
ClrScr;                                     (* очистка экрана *)
writeln('Введите: t, w, x1, b:');           (*сервисный оператор*)
read(t,w,x1,b);
writeln('Введено:');   (*контроль правильности ввода *)
writeln('t=',t:10,'  w=',w,'  x1=',x1:6:3,'  b=',b:6:3);
if (w=0.6) then s:=0 else s:=trunc((w-0.6)/abs(w-0.6)); (*вычисление ветви*)
(*ветвление*)
case s of
    -1:
       begin
       k:=1;
       f6:=sqr(t)+sin(sqr(w))/cos(sqr(w))+3;
       end;
     0:
       begin
       k:=2;
       f6:=sin(sqrt(t))+sqr(ln(x1));
       end;
     1:
       begin
       k:=3;
       f6:=exp(b*t)-cos(sqrt(t+w));
       end;
     end;
n:=2*k+6;               (*вычисление селектора*)
writeln('РЕЗУЛЬТАТ:');
writeln('N=',n:2);      (* форматный вариант вывода *)
writeln('f6=',f6:8:4);  (* форматный вариант вывода *)
end.
```

Задание. Русская пирамида - 1. Сколько кругов заданного радиуса r можно вырезать из правильного треугольника со стороной a.

При вырезании кругов наиболее эффективно используется площадь правильного треугольника при условии, что в нижнем ряду окружности вырезаются вплотную друг к другу, а во втором ряду при вырезании окружностей используются площадь, которая остается между окружностями первого ряда и т.д. Таким образом, получается правильная треугольная пирамида, в которой в каждом последующем слое количество окружностей уменьшается на единицу, пока она не завершается последним слоем, состоящим из одной окружности.

Из выше сказанного следует, что количество элементов пирамиды определяется числом окружностей, лежащих в основании пирамиды. При элементарном анализе можно получить, что в первом слое пирамиды число вырезанных из правильного треугольника стороной a окружностей с радиусом r определяется из выражения:

$$k = \text{целое}((a - 2*r*\frac{1+tg\frac{\pi}{12}}{tg\frac{\pi}{3}}) / (2*r)) = \text{целое}(\frac{a}{2r} - \frac{1+tg\frac{\pi}{12}}{tg\frac{\pi}{3}})$$

Общее число окружностей в пирамиде можно определить путем циклического суммирования количества окружностей, составляющих основание, с уменьшенным на единицу – числом окружностей во втором слое и т.д., до последнего слоя с 1 окружностью. Общее число элементов пирамиды можно также найти из следующего выражения: $n = \frac{k*(1+k)}{2}$

Псевдоалгоритм

1. Вводятся сторона треугольника и радиус окружности.
2. Определяется число окружностей в первом ряду пирамиды.
3. В случае, если число окружностей основании пирамиды меньше 1 выдается сообщение и программа завершается.
4. Вычисляется общее число окружностей.
5. Выводится результат.

Исходный текст Си-программы

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
void main (void)
{
float pi;
int   k,n;//количество окружностей
float r,//радиус окружности
      a;//сторона правильного треугольника
      pi=4*atan(1);
      printf ("Программа рассчитывает количество кругов, которые можно\n");
      printf ("вырезать из правильного треугольника\n");
      printf ("Введите радиус окружности ");
      scanf("%f",&r);
      printf ("Введите длинну стороны правильного треугольника ");
      scanf("%f",&a);
      k=int(floor(a/(2*r)-(1+tan(pi/12))/tan(pi/3)));
      if (k<1)
            {
            printf ("Невозможно вырезать окружность заданных размеров\n");
            return;
            };
      n=k*(1+k)/2;
      printf ("Можно вырезать %d окружностей радиусом %5.3f из \n",n,r);
      printf ("правильного треугольника со стороной %5.3f \n",a);
```

```
    printf ("(основание пирамиды составляют %d окружности)\n",k);
}
```

Задание. Уточните корень уравнения $e^x - 10x = 0$ на отрезке [a,b] методом хорд (с реализацией в программе ассемблерных вставок).

```
#include <iostream.h>
#include <math.h>
#pragma inline
void main(void)
{
float y,a,b,e,fa,fb,x,ten=10;
short int FPU_word;
cout<<"\nУточнения корня уравнения exp(x)-10x=0 методом хорд\n";
cout<<"Введите начало отрезка a :";      cin>>a;
cout<<"Введите конец отрезка b :";       cin>>b;
cout<<"Введите точность вычисления e :"; cin>>e;
cycle:
/*fa=exp(a)-10*a;*//*аналог кода на ассемблере*/
asm  {
FSTCW word ptr FPU_word        //чтение слова состояния FPU
OR FPU_word,0000011000000000b  //установить режим округления к нулю
FLDCW FPU_word                 //загрузка слова состояния FPU
//вычисление exp(x)=2^(x*log2(e))=(2^trunc(z))*(2^(z-trunc(z)), где z=x*log2(e)
//регистры   FPU:       |           ST(0)                 |     ST(1)       |ST(2)
//вычисляем z=x*log2(e)|                                 |                 |
//загружаем показатель |                                 |                 |
//и основание степени  |                                 |                 |
FLD  dword ptr a      //|x                                |                 |
FLDL2E                //|log2(e)                          |x                |
FMUL                  //|z                                |                 |
//вычисляем 2^z         |                                 |                 |
//создаем копию z       |                                 |                 |
FLD ST(0)             //|z                                |z                |
//округление z          |                                 |                 |
FRNDINT               //|trunc(z)                         |z                |
FSUBR ST(0),ST(1)     //|z-trunc(z)                       |z                |
F2XM1                 //|2^(z-trunc(z))-1                 |z                |
FLD1                  //|1                                |2^(z-trunc(z))-1|z
FADDP                 //|2^(z-trunc(z))                   |z                |
FSCALE                //|2^trunc(z)*2^(z-trunc(z))=exp(x) |z                |
FXCH ST(1)            //|z                                |exp(x)           |
FSTP ST(0)            //|exp(x)                           |                 |
//вычислен. exp(x)-10*x|                                 |                 |
FLD dword ptr a       //|x                                |exp(x)           |
FLD dword ptr ten     //|10                               |x                |exp(x)
FMULP                 //|10*x                             |exp(x)           |
FSUBP  ST(1),ST(0)    //|exp(x)-10*x                      |                 |
FSTP dword ptr fa     //|                                 |                 |
/*fb=exp(b)-10*b;*//*аналог кода на ассемблере*/
//регистры   FPU:       |           ST(0)                 |     ST(1)       |ST(2)
FLD  dword ptr b      //|x                                |                 |
FLDL2E                //|log2(e)                          |x                |
FMUL                  //|z                                |                 |
//вычисляем 2^z         |                                 |                 |
//создаем копию z       |                                 |                 |
FLD ST(0)             //|z                                |z                |
//округление z          |                                 |                 |
FRNDINT               //|trunc(z)                         |z                |
FSUBR ST(0),ST(1)     //|z-trunc(z)                       |z                |
F2XM1                 //|2^(z-trunc(z))-1                 |z                |
FLD1                  //|1                                |2^(z-trunc(z))-1|z
FADDP                 //|2^(z-trunc(z))                   |z                |
FSCALE                //|2^trunc(z)*2^(z-trunc(z))=exp(x) |z                |
FXCH ST(1)            //|z                                |exp(x)           |
FSTP ST(0)            //|exp(x)                           |                 |
//вычислен. exp(x)-10*x|                                 |                 |
FLD dword ptr b       //|x                                |exp(x)           |
FLD dword ptr ten     //|10                               |x                |exp(x)
FMULP                 //|10*x                             |exp(x)           |
FSUBP  ST(1),ST(0)    //|exp(x)-10*x                      |                 |
FSTP dword ptr fb     //|                                 |                 |
};
```

```
/*x=b-(a-b)*fb/(fa-fb);*//*аналог кода на ассемблере*/
asm  {
//регистры   FPU:      |           ST(0)                |     ST(1)      |ST(2)
FLD  dword ptr fb    //|fb                              |                |
FLD  dword ptr fa    //|fa                              |fb              |
FSUB ST(0),ST(1)     //|fa-fb                           |fb              |
FDIVP ST(1),ST(0)    //|fb/(fa-fb)                      |                |
FLD  dword ptr b     //|b                               |fb/(fa-fb)      |
FLD  dword ptr a     //|a                               |b               |fb/(fa-fb)
FSUB ST(0),ST(1)     //|(a-b)                           |b               |fb/(fa-fb)
FMUL ST(0),ST(2)     //|(a-b)*fb/(fa-fb)                |b               |fb/(fa-fb)
FSUBP ST(1),ST(0)    //|b-(a-b)*fb/(fa-fb)              |fb/(fa-fb)      |
FSTP dword ptr x     //|fb/(fa-fb)                      |                |
FSTP ST(0)           //|                                |                |
};
/*if (fabs(x-a)-e)<0) goto vyvod;*/ /*аналог кода на ассемблере*/
asm  {
//регистры   FPU:      |           ST(0)                |     ST(1)      |ST(2)
FLD  dword ptr x     //|x                               |                |
FLD  dword ptr a     //|a                               |x               |
FSUBP ST(1),ST(0)    //|x-a                             |                |
FABS                 //||x-a|                           |                |
FCOMP  dword ptr e   //|                                |                |
FSTSW ax             //занесение слова состояния в ax
FWAIT
SAHF                 //перенести коды условий в флаги регистра CPU
JB vyvod};
/*a=x; goto cycle;*//*аналог кода на ассемблере*/
asm  {MOV eax, x
      MOV a, eax
      JMP cycle};
vyvod:
cout<<"x="<<x<<" f(x)"<<exp(x)-10*x;
```

Задание. Экспериментально получены пять значений искомой величины функции y=f(x), найти:

1) методом наименьших квадратов эмпирическую зависимость y=f(x) в виде $y(x)=a_0+a_1x$;

2) используя функции системы MathCAD написать программу по определению эмпирических коэффициентов a_0 и a_1.

x	1	2	3	4	5
y	4.1	6.7	10.4	13	15.8

Решение

Метод наименьших квадратов

Определение коэффициентов a_0 и a_1 уравнения эмпирической зависимости методом наименьших квадратов сводится к решению следующей системы уравнений:

$$\begin{cases} a_0 n + a_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x^2{}_i = \sum_{i=1}^{n} y_i x_i \end{cases}$$, где n- количество элементов выборки (n=5)

Подставляя значения полученных экспериментальных данных, получаем следующую систему двух линейных уравнений

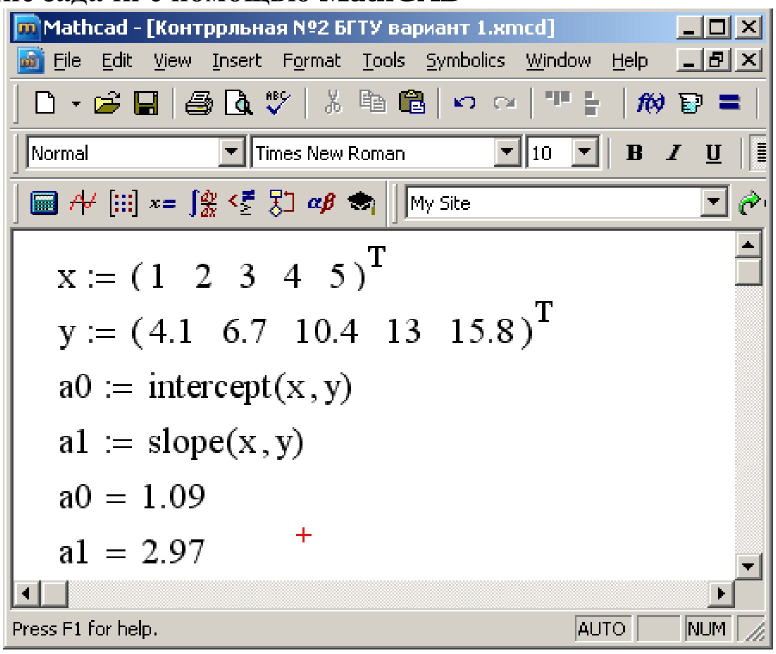
$$\begin{cases} a_0 5 + a_1(1+2+3+4+5) = 4.1+6.7+10.4+13+15.8 \\ a_0(1+2+3+4+5) + a_1(1^2+2^2+3^2+4^2+5^2) = 4.1*1+6.7*2+10.4*3+13*4+15.8*5 \end{cases}$$

Задача сводится к решению следующей простой системе линейных уравнений:

$$\begin{cases} a_0 + 3a_1 = 50 \\ 15a_0 + 55a_1 = 179{,}7 \end{cases}$$

Решение системы a_0=1,09; a_1=2,97. Таким образом, эмпирическая зависимость представляется в виде: y(x)=1.09+2.97*x.

Решение задачи с помощью MathCAD



Задание. Лесхозу требуется не более 14 трехтонных автомашин и не более 10 пятитонных автомашин. Отпускная цена автомашин: первой марки – 3000усл.ед., второй марки – 6000усл.ед. Лесхоз может выделить для приобретения автомашин от 12 до 72 тыс.усл.ед. Сколько следует приобрести автомашин каждой марки в отдельности, чтобы их суммарная грузоподъемность была максимальной. Определить максимальную суммарную грузоподъемность. Задачу решить графическим методом и написать программу для системы MathCAD по определению максимума целевой функции.

Предположим, что лесхоз приобретет x1 машину первой марки и x2 машин второй марки. Поскольку число приобретаемых машин не может быть отрицательным, а также ограничены объемы закупок каждой марки машин и выделяемые на приобретение машин средства, то должны выполняться следующие неравенства:

$3*x1+6*x2\geq12$

$3*x1+6*x2\leq72$

$x1\leq14$

$x2\leq10$

$x1,x2>0$

Общая грузоподъемность приобретаемых машин первой и второй марки составит $F=3*x1+5*x2$ (тонн).

Таким образом, приходим к следующей математической задаче: среди всех неотрицательных решений данной системы линейных неравенств требуется найти такое, при котором функция F принимает максимальное значение.

Решение графическим методом

Сначала определим многоугольник решений. Для этого в неравенствах системы ограничений и условиях неотрицательности переменных знаки неравенств заменим на знаки точных равенств и найдем соответствующие прямые:

$x1+2*x2=4$

$x1+2*x2=24$

$x1=14$

$x2=10$

$x1=0$

$x2=0$

Эти прямые изображены на рисунке 1. Каждая из построенных прямых делит плоскость на две полуплоскости. Координаты точек одной полуплоскости удовлетворяют исходному неравенству, а другой – нет. Чтобы определить искомую полуплоскость, нужно взять какую-нибудь точку, принадлежащую одной из полуплоскостей, и проверить, удовлетворяют ли ее координаты данному неравенству. Если координаты взятой точки удовлетворяют данному неравенству, то искомой является та полуплоскость, которой принадлежит эта точка, в противной случае – другая полуплоскость.

Пересечение полученных плоскостей и определяет многоугольник решений данной задачи.

Многоугольник решений

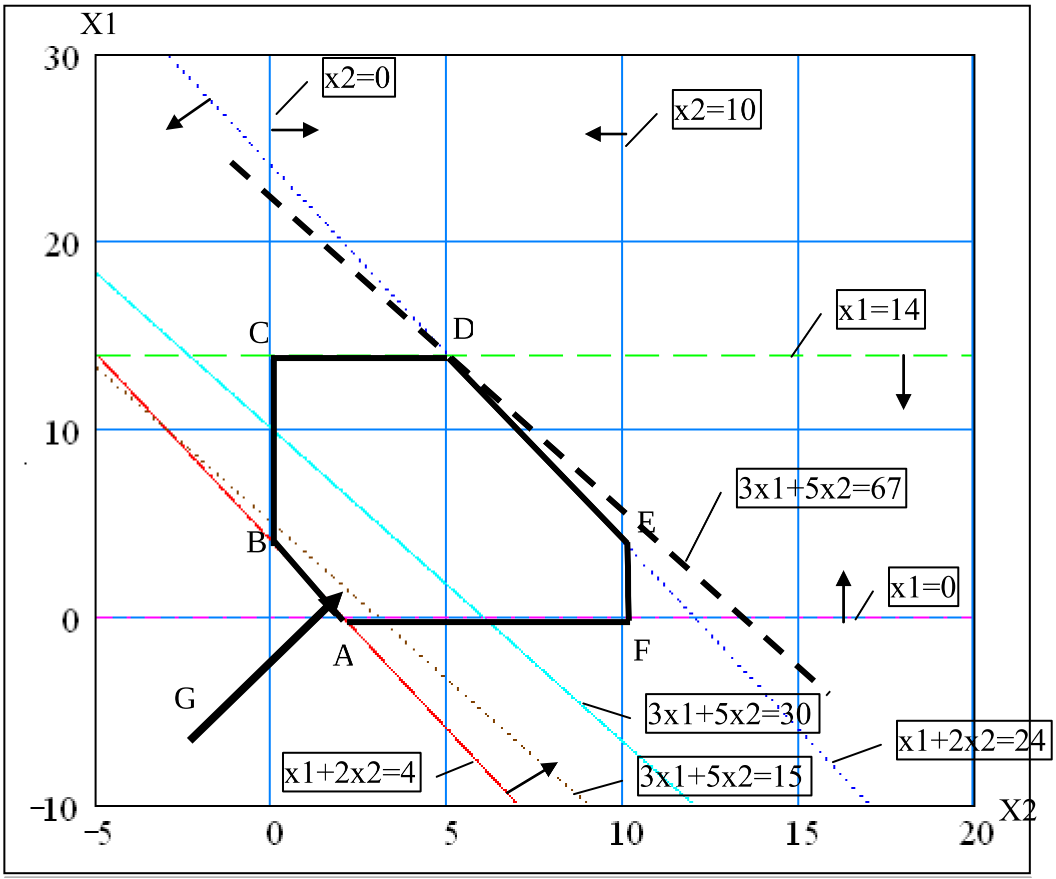


Рисунок 1.

Как видно на рисунке 1, многоугольником решений является шестиугольник ABCDEF. Координаты любой точки, принадлежащей этому шестиугольнику, удовлетворяют данной системе неравенств и условию неотрицательности переменных. Поэтому сформулированная задача будет решена, если мы сможем найти точку, принадлежащую шестиугольнику ABCDEF, в которой функция F принимает максимальное значение. Чтобы найти указанную точку, построим вектор G=(3,5) и прямую 3*x1+5*x2=h, где h – некоторая постоянная такая, что прямая 3*x1+5*x2=h имеет общие точки с многоугольником решений. Положим, например h=15 и построим прямую 3*x1+5*x2=15.

Если теперь взять какую-либо точку, принадлежащую построенной прямой и многоугольнику решений, то ее координаты определяют такой план приобретения автомашин, при котором их грузоподъемность составляет 15 тонн. Далее, полагая h равным некоторому числу, большему, чем 15, мы будем получать различные параллельные прямые. Если они имеют общие точки с многоугольником решений, то эти точки определяют планы закупки автомашин, при которых их грузоподъемность будет более 15 тонн.

Перемещая построенную прямую 3*x1+5*x2=15 в направлении вектора G, видим, что последней общей точкой ее с многоугольником решений задачи служит точка D. Координаты этой точки и определяю план закупки автомашин первой и второй марок, при котором их суммарная грузоподъемность будем максимальной.

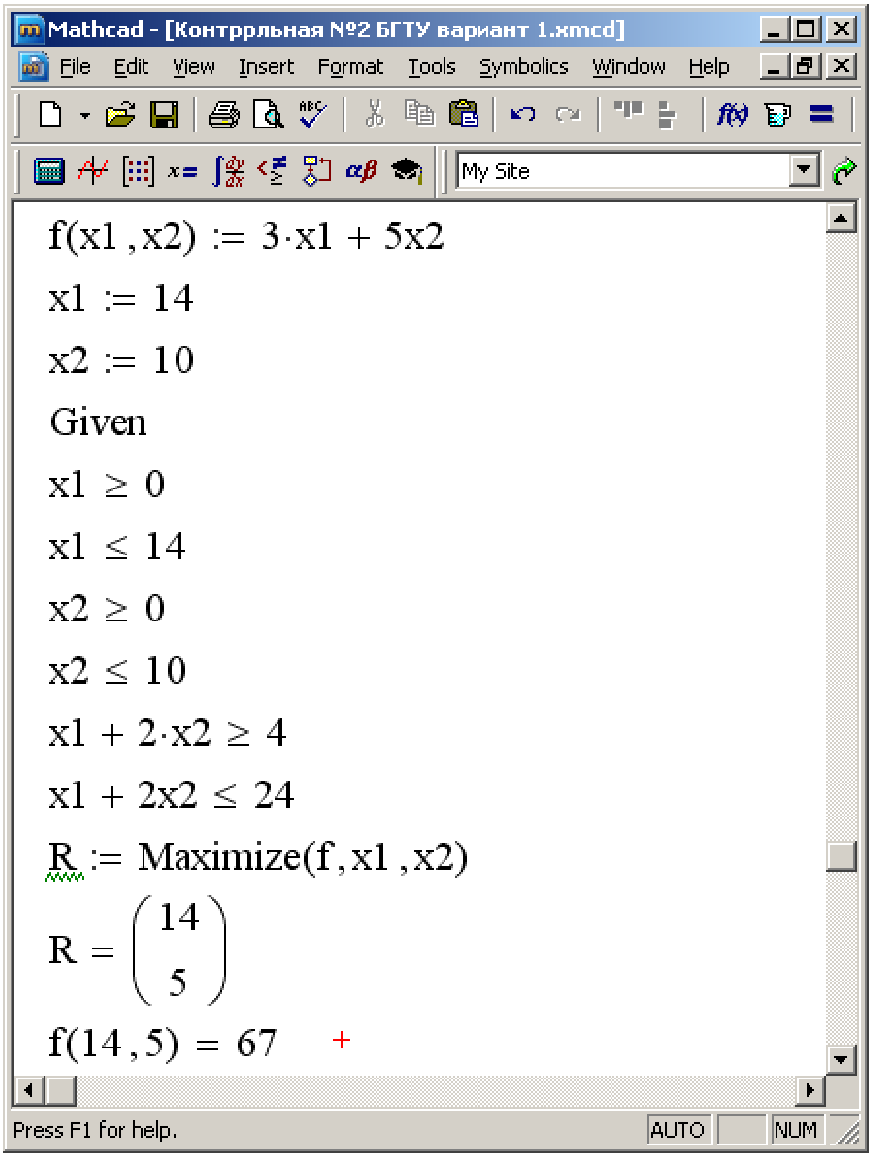
Найдем координаты точки D как точки пересечения двух прямых. Следовательно, ее координаты удовлетворяют уравнениям этих прямых:

x1=14;

x1+2*x2=24;

Получаем, x1=14, x2=5. Следовательно, лесхоз приобретет 14 трехтонных машин и 5 пятитонных машин, суммарная грузоподъемность приобретаемых автомобилей составит 3*14+5*5=67 тонн.

Решение задачи с помощью MathCAD


Mathcad - [Контррльная №2 БГТУ вариант 1.xmcd]
File Edit View Insert Format Tools Symbolics Window Help
My Site
f(x1 ,x2) := 3·x1 + 5x2
x1 := 14
x2 := 10
Given
x1 ≥ 0
x1 ≤ 14
x2 ≥ 0
x2 ≤ 10
x1 + 2·x2 ≥ 4
x1 + 2x2 ≤ 24
R := Maximize(f ,x1 ,x2)
R = (14 5)
f(14 ,5) = 67
Press F1 for help.
AUTO
NUM