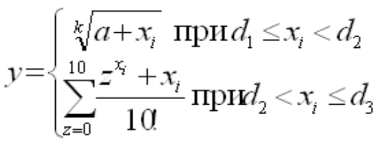
**Лабораторная работа №11**

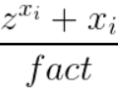
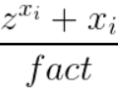
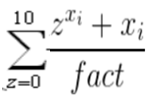
**Комбинированные вычислительные процессы.**

**Цель:** научиться комбинированным вычислительным процессам средствами компилятора free Pascal.

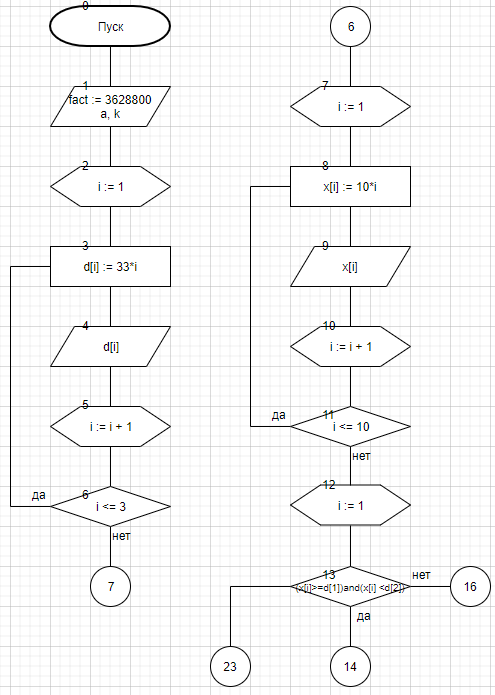
**Используемое оборудование:** пк, PascalABC.

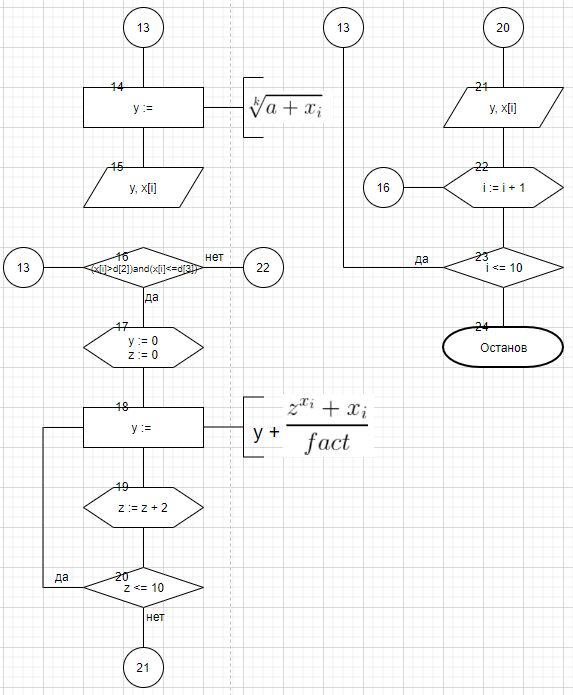
**Задача 1:**

**4.1)** Дан массив чисел X,  который состоит из элементов Xi, где i = 1 ÷ n (шаг по Z равен2) . Для элементов массива, попавших в заданный диапазон вычислить: 

**5.1)** ,, y = y + ,

**6.1)**





**7.1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | | **Тип** |
| fact | Константа, факториал числа 10 | | Int |
| y | Переменная, значение функции | | Real |
| z | Переменная, параметр цикла с шагом 2 | | integer |
| a | Переменная, аргумент | | Real |
| k | Переменная, степень корня | | Real |
| i | Переменная, параметр цикла | | Int |
| x | Массив значений x с индексом i | | Int |
| d | Массив значений границ с индексом i | Real | |

**8.1)**

**program** lab11;

**const** fact = 3628800;

**var**

z : integer;

y,a,k : real;

**var**

i : integer;

x:**array**[1..10] **of** integer;

d:**array**[1..3] **of** real;

**Begin**

writeln('Введите k: ');

readln(k);

writeln('Введите a: ');

readln(a);

writeln('d[i] : ');

**for** i:= 1 **to** 3 **do**

**begin**

d[i]:= i \* 33 ;

write( d[i], ' ');

**end**;

writeln();

writeln('x[i] : ');

**for** i := 1 **to** 10 **do**

**begin**

x[i] := i \* 10;

write( x[i], ' ');

**end**;

writeln();

**for** i:=1 **to** 10 **do**

**begin**

**if** (x[i]>=d[1]) **and** (x[i]<d[2]) **then**

**begin**

y:= power (a + x[i], 1/k);

writeln('y = ', y:2:5, ' x', i, ' = ', x[i]);

**end**;

**if** ( x[i] > d[2]) **and** (x[i] <= d[3]) **then**

**begin**

z := 0;

y := 0;

**while** z <= 10 **do**

**begin**

y := y + ((power(z, x[i]) + x[i]) / fact);

z := z + 2;

**end**;

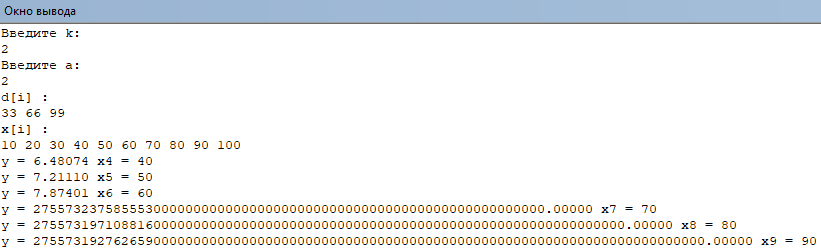
writeln('y = ', y:2:5, ' x', i, ' = ', x[i]);

**end**;

**end**;

**End**.

**9.1)**

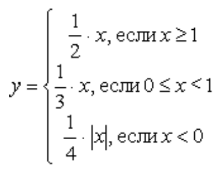


**10.1)** Задача была решена с помощью циклов конструкции for, также была использована логическая конструкция if then. Перед началом работы программы была задана константа “fact”, содержащая в себе значение факториала десяти, после задается размерность двух массивов x[10] и d[3]. Начало работы программы – запрашивается ввод переменных “k” и “a”, заполняются массивы значениями с помощью циклов конструкции for с параметром i. Следующим шагом программа попадает в цикл for c параметром i, где сразу же в логическом блоке проверяется условие (x[i]>=d[1]) **and** (x[i]<d[2]), при истинности которого функция “y” вычисляется по первому выражению, а потом выводится на экран, далее программа переходит к следующему условию, в обратном же случае программа сразу переходит к следующему условию. В этом условии проверяется ( x[i] > d[2]) **and** (x[i] <= d[3] ), если истина, то в цикле for с параметром z вычисляется функция “y” по второму выражению до тех пор, пока выполняется условие z <= 10, после вычисления функция “y” выводится на экран, а дальше параметр цикла i увеличивается на 1, иначе программа сразу переходит к увеличению параметра цикла. В конце проверяется условие цикла

i <= 10, если истина, то вычисления происходят для следующего шага, иначе конец программы.

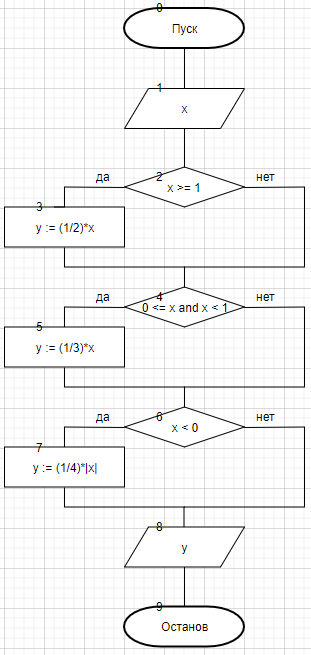
**Задача 2:**

**4.2)** Вычислить значение функции:



**5.2)** ,,

**6.2)**



**7.2)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| x | Переменная, значение x | Real |
| y | Переменная, функция | Real |

**8.2)**

**program** lab11;

**var**

x, y : real;

**Begin**

writeln ( 'Введите x : ' );

readln ( x );

**if** ( x >= 1 ) **then**

y := 1/2 \* x;

**if** ((0 <= x) **and** (x <= 1)) **then**

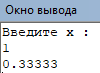
y := 1/3 \* x;

**if** ( x < 0 ) **then**

y := 1/4 \* abs(x);

writeln ( y:2:5 );

**End**.

**9.2)** 

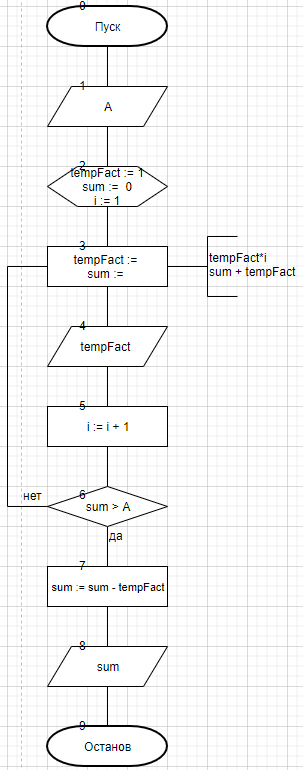
**10.2)** При решении задачи использовалась логическая конструкция If then. В начале работы программы вводится переменная “x ”. Следующим шагом программа попадает на логическое условие x >= 1, если истина, то функция “y” вычисляется по формуле 1/2 \* x, в противном случае проверяется следующее условие (0 <= x) **and** (x <= 1); при истине которого вычисление функции идет по формуле 1/3 \* x, иначе программа переходит к еще одному условию x < 0, при истине вычисления функции происходят по формуле ¼ \* x, если ложь программа переходит к выводу значения функции “y” с точностью в 5 знаков после запятой. \* Программа в любом случае проходит все условия.

**Задание 3:**

**4.3)** Составить программу подсчета суммы факториалов целых чисел, не превышающую число А, которое вводится с клавиатуры. На экран вывести сумму и все слагаемые.

**5.3)** tempFact = tempFact\*I, sum = sum + tempFact , sum = sum - tempFact

**6.3)**



**7.3)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| limA | Переменная, граница | Int |
| tempFact | Переменная, промежуточные факториалы | longInt |
| sum | Переменная, сумма факториалов и параметр цикла | longInt |
| i | Переменная, множитель | int |

**8.3)**

**program** lab11;

**var**

tempFact, limA, i, sum : integer;

tempFact, sum : longint;

**Begin**

writeln ( 'Введите A : ' );

readln ( limA );

sum := 0;

tempFact := 1;

i := 1;

write ('Слагаемые факториалы : ' );

**repeat**

tempFact := tempFact\*i;

write ( tempFact, ' ' );

sum := sum + tempFact;

i := i + 1;

**until** sum > limA;

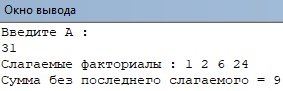
sum := sum - tempFact;

writeln ( );

writeln ('Сумма без последнего слагаемого = ', sum );

**End**.

**9.3)**



**10.3)** В ходе решения задачи использовался цикл с пост условием конструкции until repeat. С начала работы программы запрашивается переменная “limA”, после инициализируются переменные “sum”, “tempFact” и “i”. Следующим шагом программа попадает в цикл с пост условием, где вычисляется факториал, который записывается в переменную суммы “sum”, а после переменная “i” увеличивается на единицу. Проверяется условие цикла sum > limA, если ложь, то цикл продолжает работу, иначе цикл завершается и из суммы вычитается последний вычисленный факториал, после чего итоговая сумма выводится на экран.

**11)** Мы научились реализовывать алгоритмы комбинированных вычислительных процессов средствами компилятора free Pascal.