

# Комбинированные вычислительные процессы

**Цель работы:** Научиться разрабатывать алгоритмы для решения задач используя комбинированные вычислительные процессы.

**Используемое оборудование:** ПК.

## Задание №1

**Постановка задачи:** Дан массив чисел  $X$ , который состоит из элементов  $X_i$ , где  $i = 1 \div n$  (шаг по  $Z$  равен 2). Для элементов массива, попавших в заданный диапазон вычислить:

**Математическая модель:**

$$y = \begin{cases} \sqrt[k]{a + x_i} & \text{при } d_1 \leq x_i < d_2 \\ \sum_{z=0}^{10} \frac{z^{x_i} + x_i}{10} & \text{при } d_2 < x_i \leq d_3 \end{cases}$$

## Блок схема:

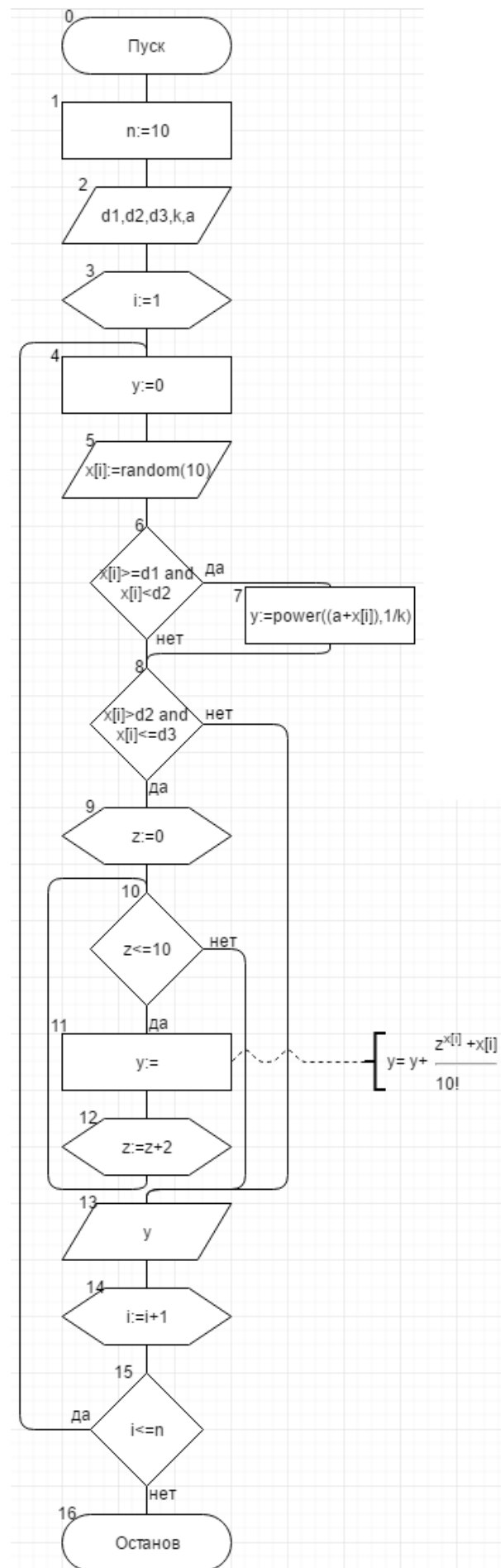


Рис 1 – блок схема к 1 задаче

## Список идентификаторов:

Таблица 1 – список идентификаторов для 1 задачи

Название	Смысл	Тип
A	Вводимое значение для расчетов	Integer
I	Параметр цикла, также адрес для вызова элемента массива	Integer
K	Вводимое значение для расчетов	Integer
Z	Параметр цикла расчета у в одном из вариантов	Integer
D1	Граница диапазона	Integer
D2	Граница диапазона	Integer
D3	Граница диапазона	Integer
Y	Результат вычислений	Real
N	Ограничение массива	Integer
X	Массив с данными для расчета	Integer

## Код программы:

```
program mda;
const n=10;

var a,i,k,z,d1,d2,d3:integer;
y:real;
x:array [1..n] of integer;

begin
  write('Введите: d1, d2, d3, k, a: ');
  readln(d1,d2,d3,k,a);
  for i:=1 to n do
    begin
      y:=0;
      x[i]:=random(10);
      if (x[i]>=d1) and (x[i]<d2) then
        y:= power((a+x[i]),1/k);
      if (x[i]>d2) and (x[i]<=d3) then
        begin
          z:=0;
          while z<=10 do
            begin
              y:=y+((power(z,x[i])+x[i])/3628800);
              z:=z+2;
            end;
          end;
          writeln('X[' ,i, ' ] = ',x[i], ' | Y[' ,i, ' ] = ',y);
        end;
    end;
  end.
```

## Результаты выполненной работы:

```
Введите: d1, d2, d3, k, a: 0 5 9 3 6
X[1] = 8 |Y[1] = 32.6616710758377
X[2] = 4 |Y[2] = 2.15443469003188
X[3] = 6 |Y[3] = 0.361826499118166
X[4] = 0 |Y[4] = 1.81712059283214
X[5] = 8 |Y[5] = 32.6616710758377
X[6] = 2 |Y[6] = 2
X[7] = 0 |Y[7] = 1.81712059283214
X[8] = 5 |Y[8] = 0
X[9] = 4 |Y[9] = 2.15443469003188
X[10] = 5 |Y[10] = 0
```

Рис 2 – результат выполнения 1 программы

**Анализ результатов вычисления:** Перед началом основного цикла программа запрашивает необходимые для работы данные, в дальнейшем все происходит в одном цикле: генерация экспериментальных переменных  $x$  и работа с ними происходит в одном цикле для ускоренной работы. Сразу же после генерации алгоритм проверяет в какой диапазон попадает  $x$  и производит с ним соответствующие операции, если же  $x$  не попадает ни в какой из диапазонов (а такое возможно, если  $x$  будет равен  $d_2$ ), то  $y$  становится равен 0, для этого и нужно обнуление  $y$  в начале цикла, ведь, если этого не будет, то там будут храниться значения предыдущих расчетов, что может повлиять на дальнейшие результаты.

## Задание №2

**Постановка задачи:** Вычислить значение функции:

**Математическая модель:**

$$y = \begin{cases} \frac{1}{2} \cdot x, & \text{если } x \geq 1 \\ \frac{1}{3} \cdot x, & \text{если } 0 \leq x < 1 \\ \frac{1}{4} \cdot |x|, & \text{если } x < 0 \end{cases}$$

**Блок схема:**

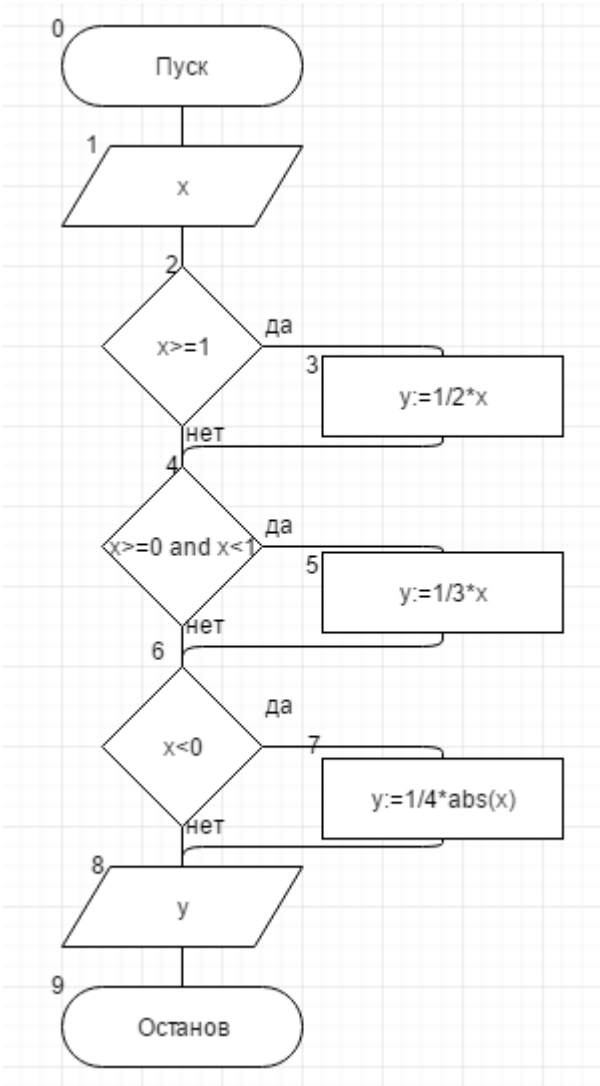


Рис 3 – блок схема к 2 задаче

**Список идентификаторов:**

Таблица 2 – список идентификаторов для 2 задачи

Название	Смысл	Тип
X	Вводимое значение	Real
Y	Результат вычислений	Real

**Код программы:**

```
program mda;

var x,y:real;
begin
  readln(x);
  if x>=1 then y:=1/2*x;
  if (x>=0) and (x<1) then y:=1/3*x;
  if x<0 then y:=1/4*abs(x);
  writeln(y);
end.
```

## Результаты выполненной работы:

4  
2

Рис 4 – результат работы 2 программы

**Анализ результатов вычисления:** Данная задача не представляет никакой сложности для тех, кто знаком с концепцией разветвляющихся вычислительных процессов. Тут необходимо написать 3 условия для расчета у в зависимости от х и в 3 условии использовать функцию `abs()` для представления х в модуле.

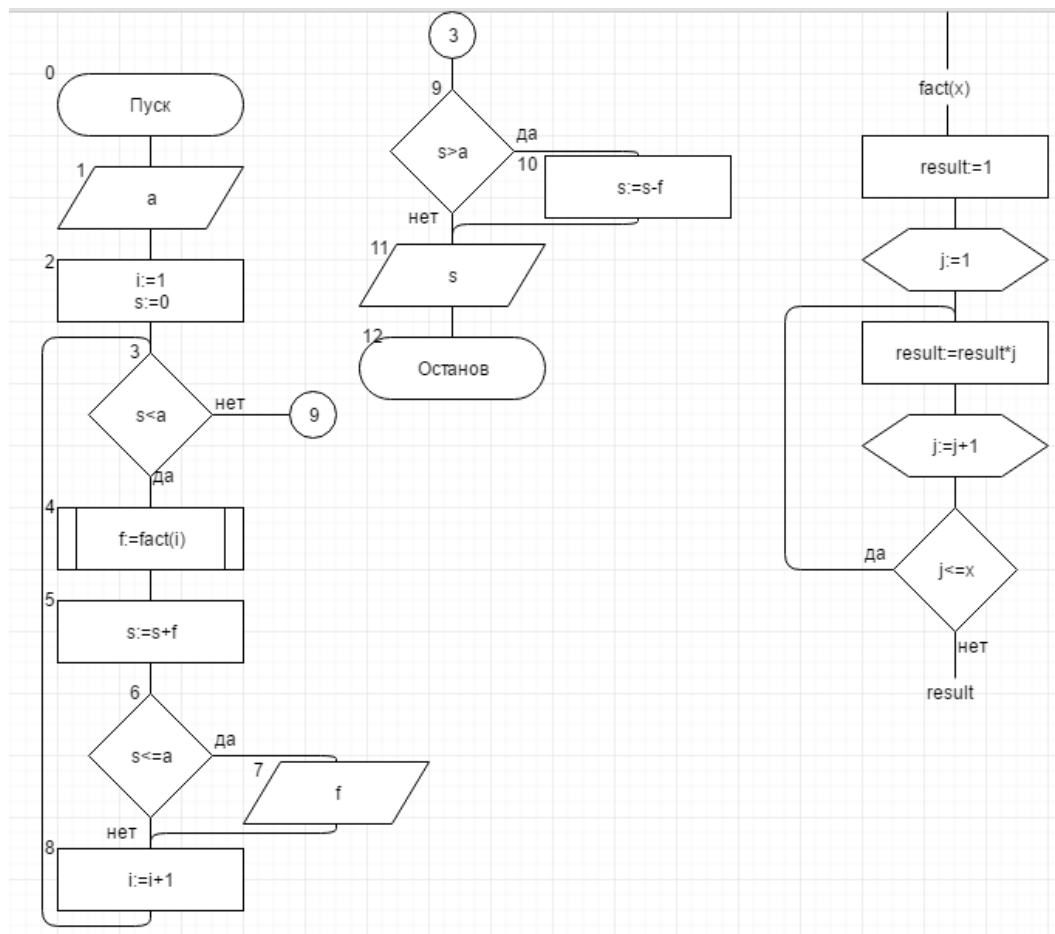
## Задание №3

**Постановка задачи:** Составить программу подсчета суммы факториалов целых чисел, где сумма не превышает число А, которое вводится с клавиатуры. На экран вывести сумму и все слагаемые.

**Математическая модель:**

$$N! = 1 * 2 * 3 * \dots * N$$

**Блок схема:**



**Список идентификаторов:**

Таблица 3 – список идентификаторов к 3 задаче

Наименование	Смысл	Тип
I	Переменная для подсчета номера выводимого слагаемого	Integer
A	Вводимая переменная	Longint
S	Сумма факториалов меньшая A	Longint
F	Переменная, хранящая в себе подсчитанный факториал очередного слагаемого	Longint
J	Параметр цикла для расчета факториала	Integer
X	Переменная функции, принимающая в себя значение передаваемое функции	Integer

**Код программы:**

```

program mda;

function fact(x:integer):longint;
var j:integer;
begin
    result:=1;
    for j:=1 to x do
        result:=result*j;
    end;

var i:integer;
a,s,f:longint;
begin
    readln(a);
    i:=1;
    s:=0;
    while s<a do
    begin
        f:=fact(i);
        s:=s+f;
        if s<=a then
        begin
            writeln('Слагаемое №',i,' : ',f);
        end;
        i:=i+1;
    end;
    if s>a then s:=s-f;
    writeln('Сумма факториалов: ',s);
end.

```

**Результаты выполненной работы:**

```

300
Слагаемое №1: 1
Слагаемое №2: 2
Слагаемое №3: 6
Слагаемое №4: 24
Слагаемое №5: 120
Сумма факториалов: 153

```

**Анализ результатов вычисления:** Задача оказалось сложнее, чем показалось на первый взгляд. Для расчета факториала каждого слагаемого была написана функция. В начале алгоритма происходит ввод  $a$ , после чего основная часть программы происходит в цикле, пока сумма факториалов  $s$  меньше  $a$ . В цикле происходит подсчет факториала  $i$ -го члена, и прибавление его к сумме, после происходит проверка меньше ли  $s$  чем  $a$ , если да, то выводится  $i$  для которого был расчет факториала, дальше  $i$  увеличивается. После выполнения тела цикла, происходит проверка, если  $s$  больше  $a$ , тогда из  $s$  вычитается последний рассчитанный факториал и результат выводится на экран.

**Вывод:** Все алгоритмы соответствуют своим поставленным требованиям, все задачи решены в соответствии с постановкой, не все алгоритмы удалось разработать с первого раза, но все задачи оказались в зоне досягаемости для решения.