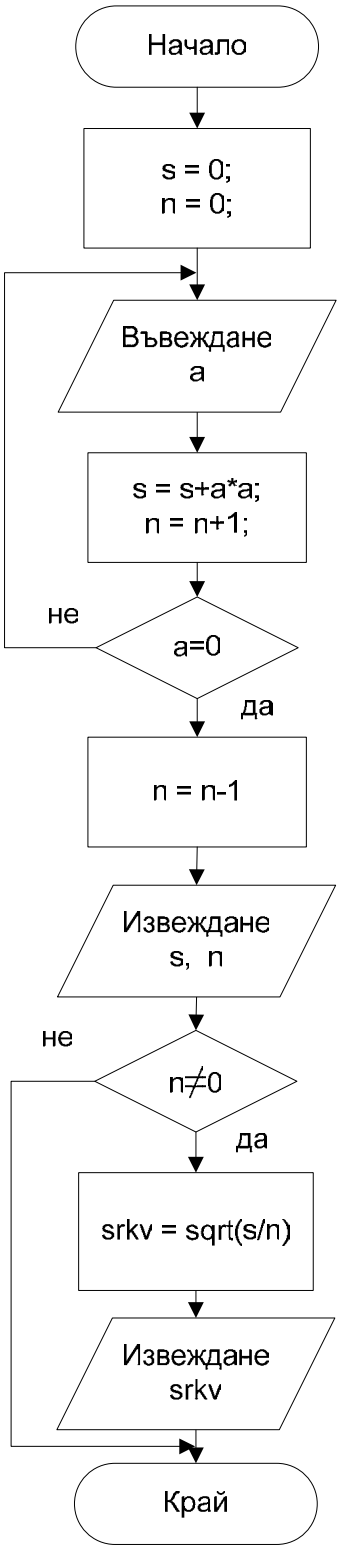
3. Циклични алгоритми и програми

Задача с_1: Да се състави програма, която въвежда последователно от клавиатурата предварително неизвестен брой числа. За край на въвеждането се задава стойност, равна на нула. Да се изчисли средноквадратичната стойност на въведените числа. В алгоритъма да се използва цикъл с постусловие (следусловие).
Средноквадратичната стойност се изчислява по формулата:

$$srkv = \sqrt{\frac{\sum_{i=1}^{n} a_i^2}{n}}$$



Решение: За да реализира въвеждане от клавиатурата и обработване на множество числа, се използва цикличен алгоритъм, който се прекратява при условие въведеното число да е равно на нула. За да се изчисли средноквадратичната стойност е необходимо първо да се намери броят и сумата от квадратите на въведените числа. След това изчислението за **srkv** може да се извърши само ако е въведено поне едно число, т.е. броят на числата **n** е различен от нула и не се допуска опит за деление на 0, което ще доведе до грешка и прекъсване изпълнението на програмата.

При реализиране на задачата с цикъл с постусловие в тялото на цикъла се преброяват всички числа, включително и въведената нула, която на практика служи за условие за край на цикъла. Това налага след излизане от цикъла броят да се намали с 1, за да се определи реалният брой на числата.

Използвани променливи величини:
srkv – средноквадратична стойност;
s – сума от квадратите на числата;
a – променлива за стойността на поредното въведено число;
n – брой на въведените числа.

1) **Програма на Pascal**

```
program c_1;
var
    srkv, s, a:real;
    n:integer;
begin
  s:=0.0;      {на сумата се задава начална стойност 0}
  n:=0;          {нулира се брояч за въведените числа}
  repeat
    write('въведи число');
    readln (a);    {въвежда се число - стойност за а}
    s:=s+sqr(a);   {добавя към сумата число на квадрат}
    n:=n+1;        {броят на числата се увеличава с 1}
  until a=0;         {условие за край на цикъла}
  n:=n-1;            {броят на числата се намалява с 1}
  writeln('броят на въведените числа е ,n:3);
  writeln('сумата от квадратите на числата е ',s:12:2)
  if n<>0 then    {ако има въведени числа, се изчислява
средноквадратичната им стойност и се извежда на екрана}
    begin
      srkv:=sqrt(s/n);
      write('средноквадратична стойност = ',srkv:12:2);
    end;
  readln;
end.
```

2) **Програма на C**

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

int main(int argc, char *argv[])
{   float a;          // въвеждано число
    double s, srkv;  // сума и средна кв.стойност
    int n;           // брой на числата
    s=0;          // начални стойности
    n=0;
    printf (" Въведи числа. За край въведи 0 \n ");
    do               // начало на циклична обработка
    {  printf(" Въведи число: ");
       scanf ("%f", &a);
       s += a*a;      // прибавя към s число на квадрат
       n++;           // броят на числата нараства с +1
    } while (a != 0);     // условие на цикъла
```

```
    n--;    // намаляване на броя заради преброената 0
    printf("Сумата от квадратите е = %12.2lf \n", s);
    printf("Броят на въведените числа е = %d \n", n);
    if (n != 0)          // ако има въведени числа
    { srkv = sqrt(s/n);     // смята средна кв.стойност
      printf("Средно квадратично = %12.2lf \n" , srkv);
    }
  system("PAUSE");
  return 0;
}
```

3) **Програма на C++**

```
#include <cmath>
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    float a;          // въвеждано число
    double s, srkv;  // сума и средно кв. стойност
    int n;           // брой на числата
    s=0;          // начални стойности
    n=0;
    cout << " Въведи числа. За край въведи 0 \n ";
    do               // начало на циклична обработка
    {   cout << " Въведи число: ";
        cin >> a;
        s += a*a;              // сумиране
        n++;                   // нарастване на брояча +1
    } while (a != 0);          // условие на цикъла
    n--;     // броят се намалява заради преброената 0
    cout << " Сумата от квадратите = " << s << endl;
    cout << " Броят на числата = " << n << endl;
    if (n != 0)          // ако има въведени числа
    {  srkv = sqrt(s/n);
       cout << " Средно квадратично = " << srkv <<
              endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

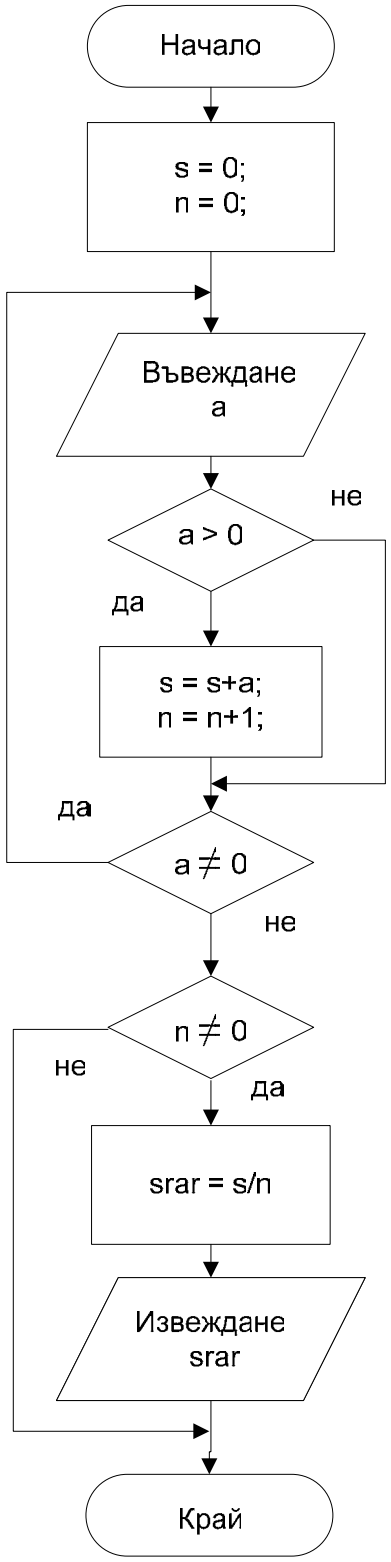
Задача с_2: Да се състави програма, с която да се въвеждат последователно от клавиатурата произволни числа, до въвеждането на стойност нула за край. Да се изчисли средноаритметичната стойност на положителните въведени числа.
Да се използва цикъл с постусловие (следусловие).
Средноаритметичната стойност се изчислява по формулата:

$$srst = \frac{\sum_{i=1}^{n} a_i}{n}$$

Решение: За да се изчисли средноаритметичната стойност първо е необходимо да се намери сумата и броят на положителните числа, които се въвеждат последователно от клавиатурата. Това означава, че всяко въведено число трябва да се провери дали е положително и само тогава да се добавя към сумата и да се преброи.

За прекратяване на цикличните действия по въвеждане и обработка на числа служи въвеждането на стойност нула.

Изчислението на средната стойност може да се извърши само ако е въведено поне едно положително число, т.е. броят на положителните числа **n** е различен от нула, за да не се допуска ситуация на опит за деление на 0.



Използвани променливи величини:
srar – средноаритметична стойност;
s – сума на положителните числа;
a – променлива за стойността на поредното въведено число;
n – брой на положителните числа.

1) **Програма на C**

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{   float  a;              // въвеждано число
    double  s=0, srar;     // сума и средна стойност
    int  n=0;             // брой на положителните числа
    do          // начало на циклична обработка
    {   printf(" Въведи число: ");
        scanf ("%f", &a);
        if (a>0)      {  s += a;
                         n++;         }
    } while (a != 0);  // край на цикъла
    if (n != 0)
    {  srar = s/n;
       printf(" Средна стойност = %.2lf \n" , srar);
    }
    else
        printf("Няма въведени положителни числа! \n");
  system("PAUSE");
  return 0;
}
```

2) **Програма на C++**

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{   double  a, s=0, srar;
    int  n=0;
    do          // начало на циклична обработка
    {   cout << " Въведи число: ";       cin >> a;
        if (a>0)      {    s += a;
                           n++;
                      }
    } while (a != 0);     // край на цикъла
    if (n != 0)
    {  srar = s/n;
       cout << " Средна стойност = " << srar << endl;
    }
    else
        cout << " Няма въведени положителни числа! \n";
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

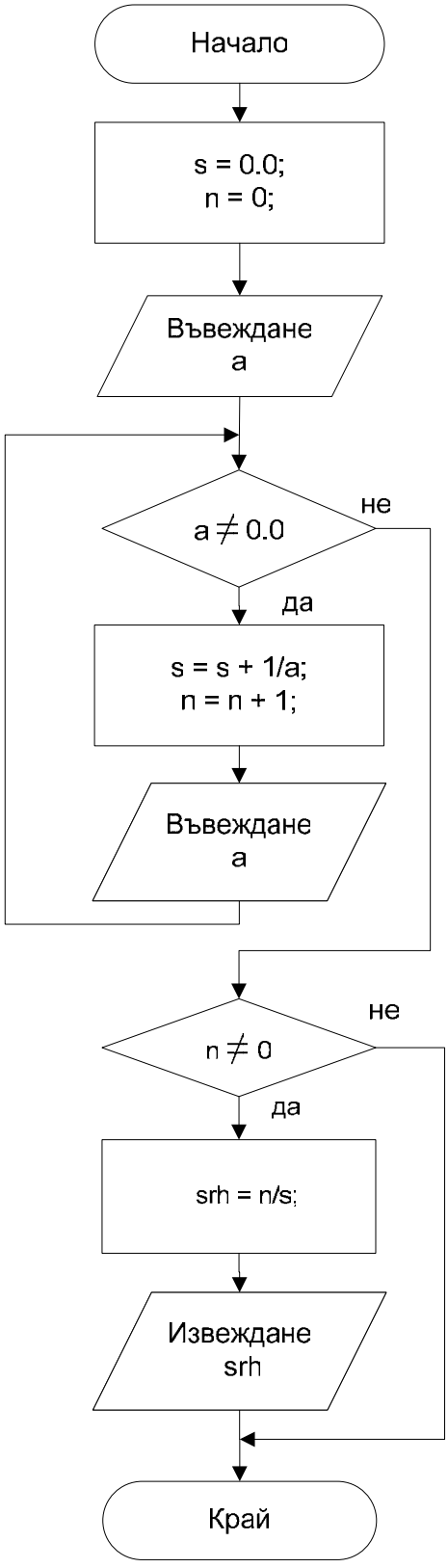
Задача с_3: Да се състави програма, с която да се въвеждат последователно произволни числа от клавиатурата до въвеждането на стойност, равна на нула. Да се изчисли среднохармоничната стойност на въведените числа, като се използва формулата:

$$srh = \frac{n}{\sum_{i=1}^{n} \frac{1}{a_i}} \quad ,$$

където n е броят на числата;
a_i - стойностите на числата.

Решение: За да се изчисли среднохармоничната стойност е необходимо да се пресметне сумата от реципрочните стойности на въведените числа и броят на тези числа **n**. Реципрочна стойност на число може да се изчисли, само ако това число е различно от нула. В противен случай при изпълнението на програмата ще има за опит за деление на 0, и изпълнението на програмата ще се прекрати. Това налага проверката за въведена стойност 0 да е преди пресмятането на реципрочната стойност на въведеното число, което на практика означава използване на цикъл с предусловие.

Изчислението се извършва само ако има въведено поне едно число, различно от нула.



Използват се следните променливи величини:

srh – среднохармонична стойност;
s – сума от реципрочните стойности на въведените числа;
a – поредното въведено число;
n – брой на въведените числа.

1) **Програма на Pascal**

```
program c_3;
var a, s, srh : real;
    n : integer;
begin
   s:=0.0;
   n:=0;
   write('въведи число');
   readln(a);
   while a<>0.0 do     {начало на цикъла с предусловие}
   begin
       s:=s+1/a;
       n:=n+1;
       write('въведи число');
       readln (a);
   end;           {край на цикъла}
   if n<>0 then  { ако има въведени числа }
   begin
      srh:=n/s;
      writeln('среднохармонична стойност е ',srh:8:2);
   end;
end.
```

2. **Програма на C**

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{   float  a;            // въвеждано число
    double  s=0, srh;    // сума и средна стойност
    int  n=0;            // брой на числата
    printf(" Въведи число: ");
    scanf ("%f", &a);
    while (a != 0)     // начало на циклична обработка
    {  s += 1/a;
       n++;
       printf(" Въведи число: ");
       scanf ("%f", &a);
    }                       // край на цикъла
    if (n != 0)             // ако има въведени числа
    {  srh = n/s;
       printf("Средна хармонична = %.4lf \n", srh);
    }
   system("PAUSE");
   return 0;
}
```

Задача с_4: Да се състави програма за изчисляване с точност **ε** на приближената стойност на функцията *y=sin x*, като се използва следната итерационна формула:
$y = \sin x = x - x^3/3! + x^5/5! - \ldots + (-1)^n x^{2n+1}/(2n+1)! + \ldots$

Решение: Въвеждаме следните помощни променливи:
a - променлива, за получаване на пореден член от реда;
i - спомагателен индекс, необходим за изчисляване на стойността на всеки следващ член на реда. Стъпката на нарастване на индекса е 2.

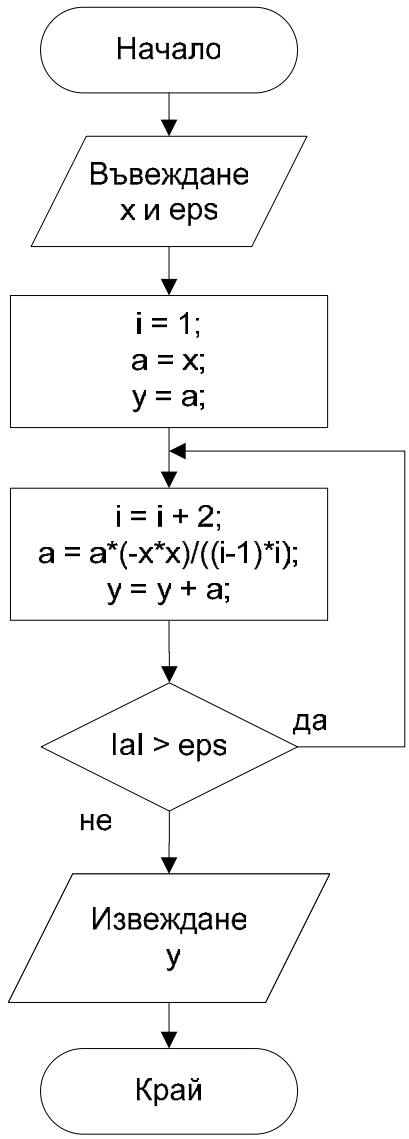
Всеки следващ член на редицата се формира от предходния член чрез умножаването му с $-x^2/((i-1)i)$, т.е.

```
a=a*(-x*x)/((i-1)*i);
```

Условие за прекратяване на изчисленията е поредният член от редицата да е по-малък от зададената точност ε. Тъй като отделните членове на редицата са с редуващи се знаци, затова е необходимо да се сравнява абсолютната им стойност $|a|<\varepsilon$

Въвежданата точност ε е число в границите от 0 до 1, например 0.001, или 0.00001.

За проверка дали полученият резултат е верен, той може да се сравни с изчислената стойност от съответната вградена функция *sin(x)*, изчислен за същата стойност на аргумента x.



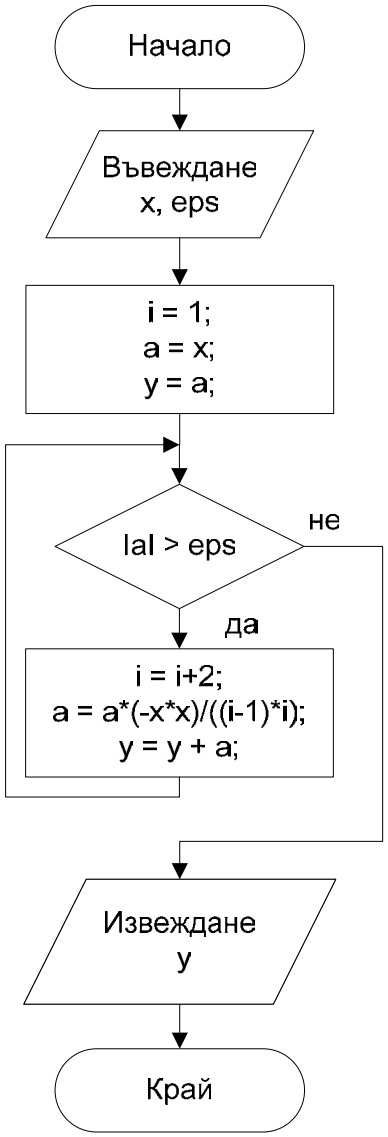
1) **Програма на C++** с цикъл със следусловие.

```
#include <cstdlib>
#include <iostream>
#include <cmath>

using namespace std;

int main(int argc, char
         *argv[])
{   double x, y, eps, a;
    int  i=1;
    cout << "x = ";
    cin >> x;
    cout << "Точност eps = ";
    cin >> eps;
```

```
    a = x;
    y = a;
    do                  // начало на циклична обработка
    {   i += 2;
        a *= -x*x/((i-1)*i);
        y += a;
    } while (fabs(a) > eps);      // край на цикъла
    cout << "Резултат y = " << y << endl;
    cout << "sin(x) = " << sin(x) << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2) **Програма на C++** с цикъл с предусловие.

```
#include <cstdlib>
#include <iostream>
#include <cmath>

using namespace std;

int main(int argc, char *argv[])
{   double x, y, eps, a;
    int i=1;
    cout << "x = ";
    cin >> x;
    cout << "Точност eps = ";
    cin >> eps;
    a = x;
    y = a;
    while (fabs(a) > eps)
    {   i += 2;
        a *= -x*x/((i-1)*i);
        y += a;
    }       // край на цикъла
    cout << "Резултат y = " << y
         << endl;
    cout << "sin(x) = " <<
         sin(x) << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```



Задача c_5: Да се изчисли стойността на израза

$Z=1/1^2 + 1/2^2 + 1/3^2 + \ldots + 1/n^2$.

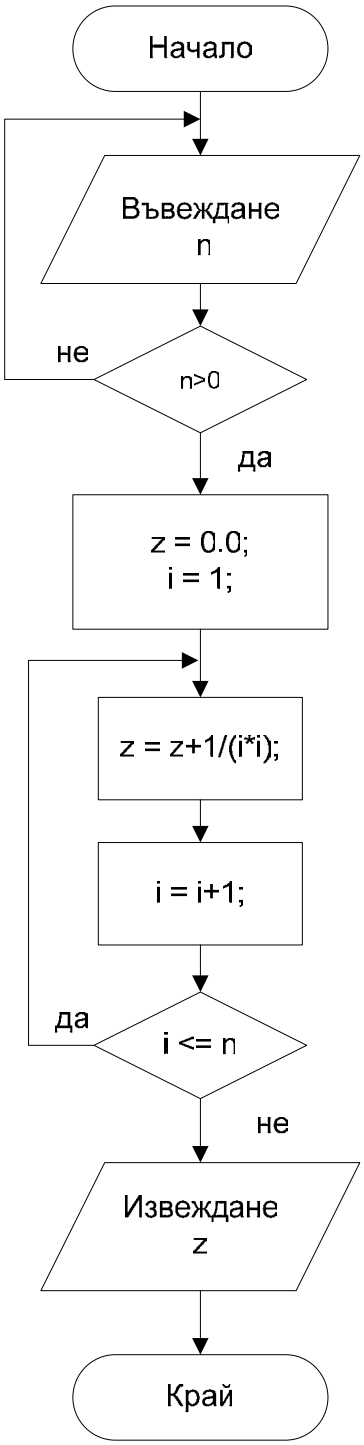
Стойността на израза се представя чрез формулата:

$$z = \sum_{i=1}^{n} \frac{1}{i^2} .$$

Решение: На практика трябва да се обработят първите **n** естествени числа. Задачата изисква първо да се зададе стойност на **n**, т.е за броя на числата, за които ще се изчислява изразът.

Сумата се формира като последователно към текущата и стойност се добавя поредното събираемо $1/i^2$ за всички стойности на **i** от 1 до **n**. Тъй като броят на изпълненията на тази обработка е известен предварително, то най-подходящо е да се използва цикъл с управляваща променлива (с брояч). Тази управляваща променлива се използва и като поредното естествено число, което се обработва.

При въвеждане на стойности от клавиатурата е препоръчително в програмата да се включва проверка дали въведените стойности са коректни, т.е дали са в допустимия диапазон, съгласно условието на задачата или техния физически смисъл. Тук **n** за „брой на числа" очевидно трябва да е положително число. Един подходящ начин за контрол на коректността на входните данни е използването на цикъл със следусловие, в който при некоректни входни данни да се повтарят действията по въвеждането им.



Използват се следните променливи:

n – брой на числата;
i – управляваща променлива за цикъла и поредно число;
z – стойност на израза.

1) **Програма на Pascal**

```
program c_5;
var i, n: integer;
    z: real;
begin
    Repeat       { цикъл за проверка на коректност }
       writeln(' въведете брой на числата n')
       readln(n);
    Until n>0;   { условие за край на цикъла}
    z:=0.0;
    for i:=1 to n do      { цикъл, с брояч от 1 до n }
        z:=z+1/sqr(i);    {тялото се изпълнява n пъти}
    write(' стойността на израза е z = ', z:10:4)
end.
```

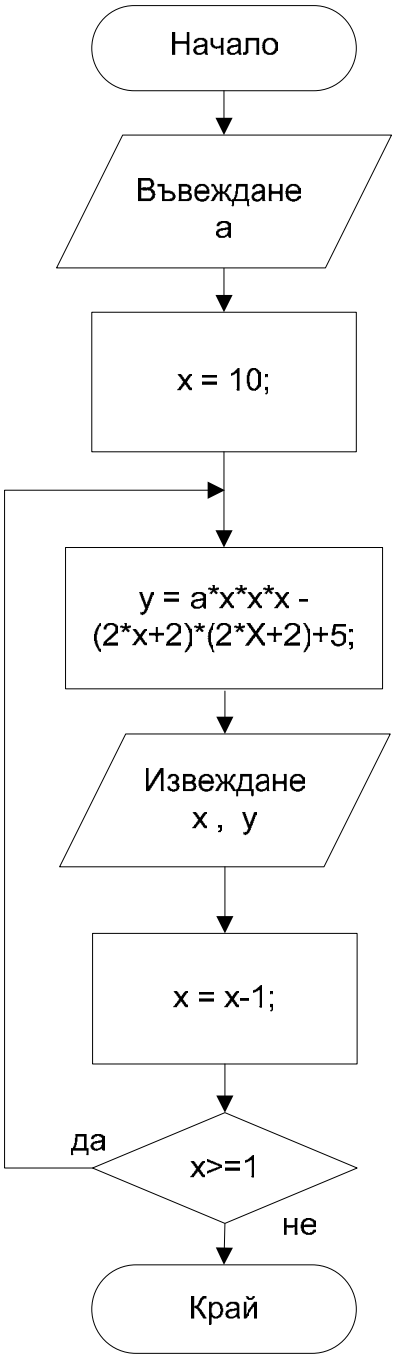
2) **Програма на C**

```
#include <stdio.h>
int main(int argc, char *argv[])
{   int  n, i;       // общ брой и брояч
    double  z=0.0;  // сума
    do          // въвеждане с проверка за коректност
    {   printf("Въведи брой на числата n= ");
        scanf ("%d", &n);
    } while (n<=1);
    for (i=1; i<=n; i++)     // цикъл с брояч
        z += 1.0/(i*i);   // 1.0 за реален тип деление
    printf("Резултат z = %lf \n" , z);
   system("PAUSE");
   return 0;
}
```

3) **Програма на C++**

```
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{   int  n, i;       // общ брой и брояч
    double  z=0.0;  // сума
    do          // въвеждане с проверка за коректност
    {   cout << " Въведи брой n= ";     cin >> n;
    } while (n<=1);
    for (i=1; i<=n; i++)     // цикъл с брояч
        z += 1.0/(i*i);    // 1.0 за реален тип деление
    cout << "Резултат z = " << z << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Задача c_6: Да се състави програма, която да изчислява стойността на функцията $y = ax^3 - (2x+2)^2 + 5$ за всички цели стойности на **x** от 10 до 1 и произволно цяло число за ***a***. Данните за **x** и **y** да се изведат в две колони.



Решение: Задача от този тип често се нарича "табулиране на функция". Според условието на задачата, изчисляването на стойността на функцията се изпълнява 10 пъти за стойности на **x** от 10 до 1. Това условие е подходящо, за да се използва цикъл с управляваща променлива, която намалява и да се използва **x** в качеството на управляваща променлива.

По условие всички данни са целочислени и аритметичните операции са с цели числа, резултатът от функцията също е цяло число.

При извеждането на резултатите, за да се подредят стойностите в две колони, се използват подходящи формати за извеждане и подходящ брой празни позиции на екрана. При език C++ за форматиране се използват манипулатори от заглавен файл <**iomanip**>. За задаване на размер на полето е **setw**(), а действието му е само за едно извеждане след него.

Използвани променливи:

a – коефициент, въвежда се;
x – аргумент на функцията;
y – стойност на функцията.

1) **Програма на Pascal**

```
program c_6;
var   a, x, y: integer;
begin
  write('въведете стойност за коефициента a');
  readln(a);
```

```
  writeln('табулация на функция');
  writeln(' ':3,'x',' ':8,'y');   {извежда се заглавие}
  for x:=10 downto 1 do       {цикъл с намаляващ брояч}
   begin
    y:=a*x*x*x-sqr(2*x+2)+5; {пресмятане на функцията }
    writeln(x:6, y:9);
   end;                        {край на цикъла}
end.
```

2) **Програма на C**

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{   int  a, x, y;
    printf ("Въведи a = ");
    scanf ("%d", &a);
    printf ("\n Табулиране на функция \n");
    printf ("         x          y  \n\n");
    for (x=10; x>=1; x--)   // цикъл с намаляващ брояч
    {    y = a*x*x*x - (2*x+2)*(2*x+2) + 5;
         printf ("%10d %10d \n", x, y);
    }
  system("PAUSE");
  return 0;
}
```

3) **Програма на C++**

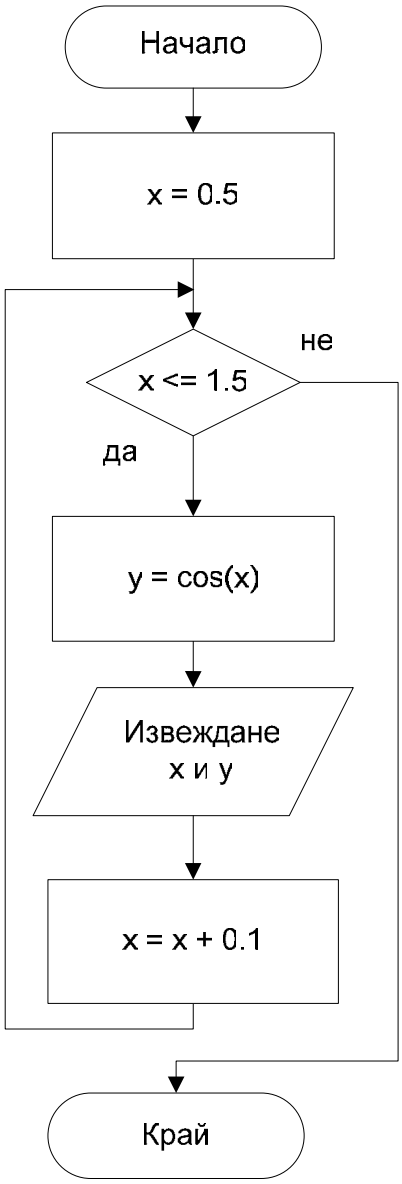
```
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[])
{   int  a, x, y;
    cout << "Въведи a = ";
    cin >> a;
    cout << "\n Табулиране на функция \n";
    cout << "         x          y  \n\n";
    for (x=10; x>=1; x--)   // цикъл с намаляващ брояч
    {  y = a*x*x*x - (2*x+2)*(2*x+2) + 5;
       cout << setw(10) << x << setw(10) << y << endl;
    }     // setw(10) за размер на полето за извеждане
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Задача с_7: Да се състави програма, която да изчислява стойността на функцията y=cos(x) за стойности на аргумента x=0.5, 0.6, 0.7,…,1.4, 1.5 rad.
Да се изведат всички стойности на x и y.

Решение: Според условието на задачата стойностите за аргумента са реални числа. В език Pascal, ако за такъв случай се използва цикъл с целочислен брояч, то е необходимо отделно да се изменя и стойността на аргумента **x** или да се намери подходяща зависимост за определяне на стойността на **x** за всяка стъпка от изпълнението на цикъла с брояч.

При език C/C++ цикълът с управляваща променлива може да се използва с реални числа и броячът да се изменя с реална стъпка 0.1.

За форматирано извеждане в C++ се използват манипулатори **setw()**и **setprecision()** от файл <**iomanip**>.



Използвани променливи:
x – аргумент на функцията;
y – стойност на функцията;

Програма на C++

```
#include <cstdlib>
#include <cmath>
#include <iostream>
#include <iomanip>
using namespace std;
int main(int argc, char *argv[])
{   double  x, y;
    cout << "\n Табулиране на функция y=cos(x) \n";
    cout << "\t x  \t  y \n\n";
    cout << setprecision(4);  // брой десетични знаци 4
    for (x=0.5; x<=1.5; x+=0.1)  /* цикъл с намаляваща
              управляваща променлива с реално число */
    {    y = cos(x);
         cout << setw(8) << x << setw(10) << y << endl;
    }
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Задача c_8: Да се състави програма, която да изчислява стойността на израза:

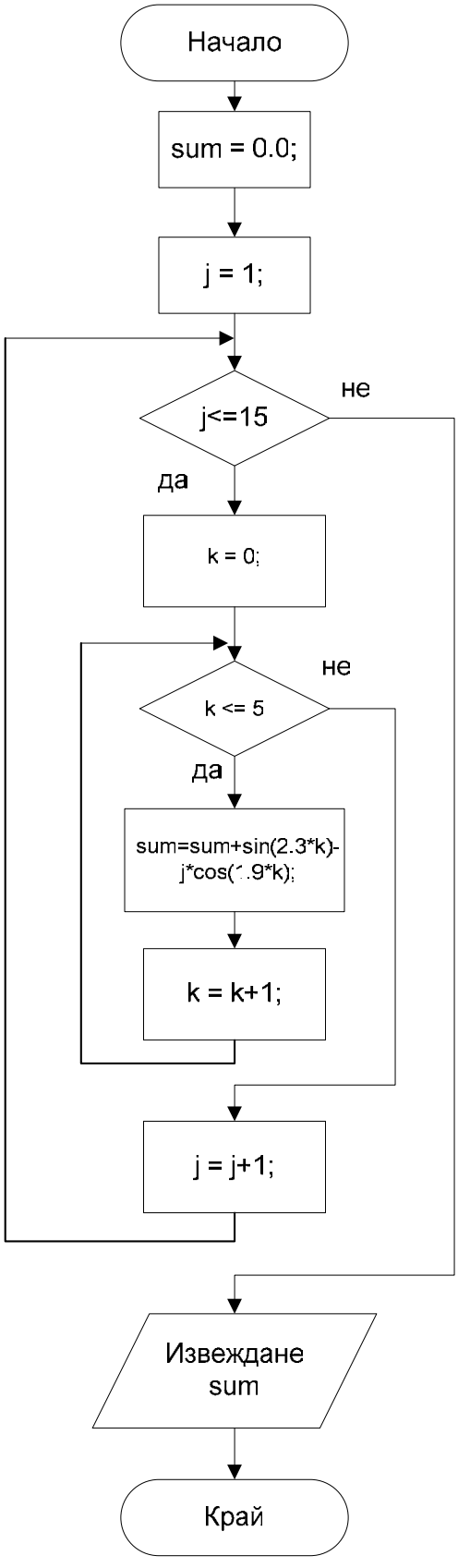
$$s = \sum_{j=1}^{15} \sum_{k=0}^{5} (\sin 2.3k - j^2 \cos 1.9k)$$

Решение: В израза са включени две вложени суми и за изчисляването им е необходимо да се използват два вложени оператора за цикъл с брояч.
В език C/C++ цикъл for е реализиран с предусловие.

Използвани променливи:

j и **k** – аргументи на функцията и управляващи променливи за циклите;

sum – стойност на израза.



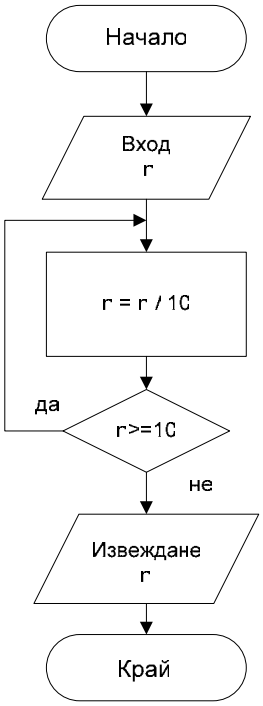
Програма на C++

```
#include <cstdlib>
#include <iostream>
#include <cmath>
using namespace std;
int main(int argc, char
              *argv[])
{  int j, k;
   double sum=0.0;
   for (j=1; j<=15; j++)
     for (k=0; k<=5; k++)
     {  sum += sin(2.3*k) -
            j*j*cos(1.9*k);
     }
   cout << "Sum = " << sum <<
           endl;
   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Идея: За да се провери как се изменя стойността на сумата, може извеждането на резултата да се постави вътре в тялото на циклите, т.е. вътре в скобите { }.

Задачи за самостоятелна работа:

1. Да се напише програма за намиране сумата на първите N естествени числа кратни на 3.
2. Да се напише програма за намиране произведението на първите N естествени числа (факториел N!)
3. Да се напише вариант на програмата с_5 като се използва цикъл с предусловие за проверка на входните данни.
4. Да се напише програма, която намира следната сума: $y = \cos x + \cos 2x + \cos 3x + ... + \cos 9x$ при еднократно въведена стойност за ъгъла x. Да се направят варианти с използване на трите оператора за цикъл.
5. Да се напише програма за изчисляване на функцията $y = 3\sqrt{x+2} + 3.9x^2$, за всички цели стойности на x в интервала $x \in [2,20]$.
6. Да се напише програма, която въвежда последователно от клавиатурата предварително неизвестен брой числа до въвеждането на стойност нула. Да се изчисли средноаритметичната стойност на тези от числата, които са в интервала (3,6).
7. Да се напише програма, която въвежда последователно от клавиатурата предварително неизвестен брой числа до въвеждането на стойност нула. Да се изчисли произведението на тези числа, които са кратни на 5.
8. Да се напише програма, която намира първата цифра на дадено естествено число N (при N>10).



Съвети:

Съобразете какъв тип данни е най-подходящо да се използва.

За задача 8 използвайте последователно многократно целочислено деление. Един вариант на решение може да е следният алгоритъм:

Направете и вариант на решение с цикъл с предусловие.

Можете да добавите и проверка за коректността на въведеното число за N в задачи 1,2 и 8.