Лабораторно упражнение N 6
Итерационни конструкции в програмен език С

I. Теоретична обосновка

Циклични алгоритми са тези при които има многократно повторение на група действия. Многократното повторение на група действия се нарича ***цикъл***. Конструкциите за цикли в С са част от управляващите, наричат се още **итерационни**. Предвидени са три итерационни конструкции, които реализират цикли в програмите: while, do-while, for. Тялото на цикъла се състои от един оператор. Ако алгоритъмът изисква в тялото да са повече от един оператори, те се обединяват чрез съставният оператор **{…}**.

1. Конструкция while

Чрез конструкция while се реализира цикъл с пред условие.Синтаксисът е следният:

while (условие) оператор;

Действието е следното: Проверява се условието. Ако стойността на израза е различна от 0 (TRUE), изпълнява се тялото на цикъла и отново се проверява условието за край, т.е.оператора в тялото за цикъла се изпълнява, докато условието е вярно. Когато стойността на условието стане равна на 0 (FALSE), излиза се от цикъла.

2. Конструкция за цикъл със след условие do-while

Конструкция do-while реализира цикъл със след условие. Синтаксисът е следния:

do оператор;
while (условие);

Действието е следното: Операторът в тялото на цикъла се изпълнява до тогава докато условието е вярно т.е изразът представящ условието има стойност различна от 0 (TRUE). Когато стойността на израза стане равна на 0 (FALSE), прекратява се изпълнението на цикъла.

3. Конструкция for

Итерационна конструкция for се използва предимно за реализация на цикли с известен брой повторения. По своето действие, конструкцията реализира цикъл с предусловие.

Синтаксисът е следния:

for (секция 1;секция 2;секция 3) оператор;

Секция 1 и секция 3 се състоят от един или няколко оператора, отделени със запетая. Секция 2 е условен израз.

Действието е следното: Изпълняват се операторите в секция 1. Това е т.нар. ***инициализираща секция***, която се изпълнява еднократно и с която се задават началните условия на цикъла. Изпълнението на for продължава в следната последователност: изчисляване на секция 2; изпълнение на оператора след скобите; изпълнение на секция 3. Тази последователност се изпълнява дотогава, докато стойността на израза в секция 2 има стойност различна от 0 (TRUE).

4. Примери за реализация на циклични алгоритми

4.1. Намиране на N факториел

```
// пример 1- намиране на N факториел чрез използване на оператор while
#include <iostream.h>

main()
{ int i;
  unsigned n,nf;

  cout<<" Input n=";
  cin>>n;

  nf=1;
  i=1;
  while(i<=n)
   { nf*=i;
     i++;
   }
 cout<<"\n N facturiel = "<<nf;
 return 0;
}

// пример 2: намиране на N факториел чрез използване на оператор do-while
#include <iostream.h>

main()
{ int i;
  unsigned n,nf;

  cout<<" Input n=";
  cin>>n;
  nf=1;
  i=1;
  do
   { nf*=i;
     i++;
   }  while(i<=n);
 cout<<"\n N facturiel = "<<nf;
 return 0;
}

// пример 3: намиране на N! чрез използване на оператор for

#include <iostream.h>

main()
{ int i;
  unsigned n,nf;
```

```
 cout<<" Input n=";
 cin>>n;

 nf=1;
 for(i=1;i<=n;i++) nf*=i;
 cout<<"\n N facturiel = "<<nf;
 return 0;
}
```

4.2. Намиране на средноаритметична стойност

Задачата е за намиране на средноаритметична стойност от произволен брой положителни числа, въведени от клавиатурата.

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

main ()
{ float x,sum,srst;
  int k;
  k=0;
  sum=0;
 do
   {  printf ("krai na vavejdaneto e 0\n");
      printf ("vavedi  x na broi polojitelni chisla:\n");
       scanf ("%f",&x);

      if (x>0)
        { k++;
          sum+=x;
        }
  }  while(x!=0);
 srst=sum/k;
 printf ("srst=%f",srst);
 return 0;
}
```

III. Задачи за изпълнение

1. Да се създаде конзолно приложение, което намира сумата на произволно въведени положителни числа от клавиатурата. Въвеждането на стойност 0 да прекрати понататъшното въвеждане на числа.

2. Да модифицира програмния код в задача 1, така че приложението да намира произведението на произволно въведени положителни числа от клавиатурата.

3. Да се създаде конзолно приложение, с което се извеждат на екрана стойностите от 1 до n в прав и обратен ред, като стойността на n се въвежда от клавиатурата.

4. Да се създаде конзолно приложение, което отпечатва на екрана следното:

```
1
1 2
1 2 3
1 2 3 4
```

…
1 2 3 4 … n
Стойността на n се въвежда от клавиатурата.

5. Да се създаде конзолно приложение, което отпечатва на екрана следното:

```
      0
     101
    21012
   3210123
n..3210123..n
```

Стойността на n се въвежда от клавиатурата.

6. Да се състави алгоритъм и програма за намиране на e^x в ред на Фурие. Натрупването на междинната сума да се прекрати, когато поредния член стане по-малък от 10^{-8}. Формулата за пресмятане е: $e^x=1+x/1+x^2/2!+x^3/3!+...+x^n/x!$

Да се сравни получената стойност със стойността, получена чрез използване на функция exp().