Упражнение №5

ОПЕРАТОРИ ЗА ЦИКЪЛ `while`, `do-while` и `for`. ПОТОКОВИ ЗАДАЧИ В С

Цикълът е базова управляваща структура за многократно изпълнение на инструкциите в даден блок. Основните елементи на цикъла са:

- *задаване на начално състояние – инициализация;*
- *условие за продължение;*
- *блок от инструкции, който се изпълнява многократно;*
- *модификация – част от блока,* в *която се променят данните, които участват* в *условието за продължение, така* че *на определен етап да може да се излезе от цикъла.*

В зависимост от мястото, където е разположено условието за продължение, циклите се делят на: цикъл с пред-условие – в езика С се реализира с операторите **`while`** и **`for`**, и цикъл с пост-условие – в С се реализира с оператора **`do-while`**. При програмирането на обработката в цикъла често се използват операторите **`break`** – за напускане на цикъла и **`continue`** – за приключване на текущата итерация и продължение на цикъла.

Синтаксис:

```
while(израз)
      оператор

do
      оператор
while(израз);
      for(инициализация; израз; актуализация)
оператор
```

Пример 1: Да се изчисли сумата на целите числа в зададен диапазон [a, b] – използване на цикъл с пред-условие – оператор **`while`**.

```
#include <stdio.h>
int main()
{
  int a,b,i;
  int sum;

  printf("Vyvedi a: ");
  scanf("%d", &a);
  printf("Vyvedi b: ");
  scanf("%d", &b);

  sum=0;
  i=a;
  while(i<=b)
  {
    sum+=i;
    i++;
  }

  printf("Sumata mejdu %d i %d = %d\n", a, b, sum);
  return 0;
}
```

Същата обработка, но с използване на оператор **for** и операция запетая:

```
#include <stdio.h>
int main()
{
  int a,b,i;
  int sum;
  printf("Vyvedi a: ");
  scanf("%d", &a);
  printf("Vyvedi b: ");
  scanf("%d", &b);
  for(sum=0, i=a; i<=b; sum+=i, i++);
  printf("Sumata mejdu %d i %d = %d\n", a, b, sum);
  return 0;
}
```

Пример 2: Да се организира проверка за коректност на въвеждани данни – например дали стойността на реална променлива **eps** лежи в диапазона (0, 0.1] – използване на цикъл с пост-условие (оператор **do-while**).

```
#include <stdio.h>
int main()
{
  double eps;
  do
  {
    printf("Vyvedi eps: ");
    scanf("%lf", &eps);
   }
  while(!(eps>0 && eps<=0.1));
  printf("eps = %lf \n", eps);
  return 0;
}
```

Пример 3: Определяне на броя на цифрите в последователност от символи – използване на оператор **continue**.

```
#include <stdio.h>
int main()
{
  char c;
  int i, br, total;

  printf("Vyvedi obsht broj na simvolite: ");
  scanf("%d", &total);          // Общ брой на символите

  br=0;                         // Брой на цифрите
  for(i=0; i<total; i++)
  {
    c=getchar();
    if(c<'0' || c>'9')
     continue;
    br++;
  }

  printf("Broj na cifrite = %d\n", br);
  return 0;
}
```

Пример 4: Изчисляване в цикъл на корен квадратен от въведено число, прекратяване на цикъла с въвеждане на отрицателна стойност – използване на оператор **break**.

```
#include <stdio.h>
#include <math.h>
int main()
{
  double x;
  while(1)
  {
    printf("Vyvedi x: ");
    scanf("%lf", &x);
    if(x<0.0)
     break;
    printf("Koren kvadraten ot %lf = %lf\n", x, sqrt(x) );
  }
  return 0;
}
```

Потокът може да се разглежда като поредица от еднотипни елементи, които се получават и обработват последователно – един по един. Във всеки момент е достъпен за обработка само един – текущият елемент. При въвеждане на следващ елемент новата стойност се поставя на мястото на предишната и последната не е повече достъпна. При въвеждане на елементите от потока по принцип е необходима памет за запазване само за текущия елемент. Обработката представлява обновяване на резултата въз основа на текущата стойност и не изисква съхранението на всички елементи от поредицата в оперативната памет.

Обща схема на обработка елементите на поток:

```
Задаване на начално състояние
Въвеждане на стойност на първия елемент
Докато ( ! Условие за спиране)
{
 Обработка
 Въвеждане на стойност на следващ елемент
}
```

Условие за спиране на обработката на елементите на входния поток:

- Достигане на предварително известен брой елементи
- Въвеждане на „специална (ограничителна) стойност“
- Въвеждане на край на файла (EOF)

Видове обработка – определяне на:

- Сума
- Брой
- Средна стойност
  с/без допълнително условие
  Максимален елемент (и неговият пореден номер)
- Минимален елемент (и неговият пореден номер)
- Настъпване на определено събитие – вдигане на флаг

Пример 5: Изчисляване на сумата на положителните елементи и определяне на броя на отрицателните елементи, обработката продължава довъвеждане на стойност -999.

```
#include <stdio.h>
int main()
{
  int a, sum, broi;
```

```
  sum=0;
  broi=0;
  printf("Vyvedi a: ");
  scanf("%d", &a);
  while (a != -999)              // -999 е специална (ограничителна) стойност
  {
     if (a>=0)
        sum += a;
     else
        broi++;
     printf("Vyvedi a: ");
     scanf("%d", &a);
  }
  printf("sum = %d, broi = %d\n", sum, broi);
  return 0;
}
```

Пример 6: Изчисляване на сумата на всичките елементи със спиране на обработката при въвеждане на край на файла.

```
#include <stdio.h>
int main()
{
  int a;
  int i;
  int sum;
  sum=0;
  i=0;
  printf("Vyvedi a: ");
  while (scanf("%d", &a) != EOF)       // край на файл (EOF)
  {
    i++;
    sum += a;
    printf(" Stojnost na i= %d, Stojnost na sum= %d\n", i, sum);
    printf(" Vyvedi a za sledvashta iteracia: ");
  }
  printf("sum = %d\n", sum);
  return 0;
}
```

```
Vyvedi a: 3
 Stojnost na i= 1, Stojnost na sum= 3
 Vyvedi a za sledvashta iteracia: 4
 Stojnost na i= 2, Stojnost na sum= 7
 Vyvedi a za sledvashta iteracia: 5
 Stojnost na i= 3, Stojnost na sum= 12
 Vyvedi a za sledvashta iteracia: 6
 Stojnost na i= 4, Stojnost na sum= 18
 Vyvedi a za sledvashta iteracia: ^Z
sum = 18
Press any key to continue
```

EOF

Пример 7: Намиране на стойността на максималния елемент и на неговия пореден номер, спиране на обработката при въвеждане на край на файла:

```
#include <stdio.h>
int main()
{
  int a, max, i, number;
```

```
  i=1;
  printf("Vyvedi pyrva stojnost na a: ");
  scanf("%d", &a);
  max = a; number=i;
  printf(" Vyvedi sledvashta stojnost na a: ");
  while (scanf("%d", &a) != EOF)            // край на файл (EOF)
  {
    i++;
    if(a > max)
    {
          max=a; number = i;
    }
    printf(" Stojnost na i= %d, Stojnost na max= %d\n", i, max);
    printf("Vyvedi sledvashta stojnost na a: ");
  }
  printf("max = %d, number = %d\n", max, number);
  return 0;
}
```

Пример 8: **Следене за настъпване на определено събитие – вдигане на флаг.** Да се следи дали сред въвежданите елементи има кратни на 7 – при откриване на такъв елемент да се вдигне флаг и да се запомни поредният му номер.

```
#include <stdio.h>
int main()
{
  int a, i, flag, number;
  flag=0;                                // неуспешна операция
  i=0;
  printf("Vyvedi a: ");
  while (scanf("%d", &a) != EOF)         // край на файл (EOF)
  {
    i++;
    if (a%7 == 0)                        // или if ( !(a%7))
    {
      flag = 1;                          // успешна операция
      number = i;
    }
    printf(" Stojnost na i= %d, Stojnost na flag= %d\n", i, flag);
    printf(" Vyvedi a za sledvashta iteracia: ");

  }
  if (flag)
     printf("Ima kratni elementi - posledniqt e s nomer = %d\n", number);
  else
     printf("Njama kratni elementi.\n");
  return 0;
}
```