

Упражнение №8

ОПЕРАТОРИ ЗА ЦИКЪЛ `while`, `do-while` и `for`

ПОТОКОВИ ЗАДАЧИ В C

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 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 1: Изчисляване на сумата на положителните елементи и определяне на броя на отрицателните елементи, обработката продължава до въвеждане на стойност -999.

```
#include <stdio.h>
int main()
{ int a, sum, broi;
  sum=0;
  broi=0;
  printf("Input a: ");
  scanf("%d", &a);
  while (a != -999)
  {
    if (a>=0)
      sum += a;
    else
      broi++;
    printf("Input a: ");
    scanf("%d", &a);
  }
}
```

```

    printf("sum = %d, broi = %d\n", sum, broi);
    return 0;
}

```

Пример 2: 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("Input a: ");
    while (scanf("%d", &a) != EOF)
    {
        i++;
        sum += a;
        printf(" Value of i= %d, Value of sum= %d\n", i, sum);
        printf("    Input a for next iteration: ");
    }
    printf("sum = %d\n", sum);
    return 0;
}

```

```

Input a: 3
Value of i= 1, Value of sum= 3
    Input a for next iteration: 4
Value of i= 2, Value of sum= 7
    Input a for next iteration: 5
Value of i= 3, Value of sum= 12
    Input a for next iteration: 6
Value of i= 4, Value of sum= 18
    Input a for next iteration: ^Z
sum = 18
Press any key to continue

```

EOF

Пример 3: 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("Input first value of a: ");
  scanf("%d", &a);
  max = a; number=i;
  printf("    Input next value: ");
  while (scanf("%d", &a) != EOF)
  { i++;
    if(a > max)
    {
        max=a; number = i;
    }
    printf(" Value of i= %d, Value of max= %d\n", i, max);
    printf("    Input next value: ");
  }
  printf("max = %d, number = %d\n", max, number);
  return 0;
}

```

Пример 4: 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;
    int i;
    int flag;
    int number;
    flag=0;
    i=0;
    printf("Input a: ");
    while (scanf("%d", &a) != EOF)
    {
        i++;
        if (a%7 == 0)                // или if ( !(a%7) )
        {                            //      {
            flag = 1; number = i;    //      flag = 1; number = i;
        }                          //      }
        printf(" Value of i= %d, Value of flag= %d\n", i, flag);
        printf("    Input a for next iteration: ");
    }
    if (flag)
        printf("Ima kratni elementi - posledniqt e s nomer = %d\n", number);
    else
        printf("Njama kratni elementi.\n");
    return 0;
}
```