

# Цикъл с постусловие



цикъл do ... while

# Общ вид на командата

**do**

*команда;*

**while** (*условие*);

**do**

{

*команда 1;*

*...*

*команда N;*

}

**while** (*условие*);

□ **команда** –

произволна команда  
от езика C / C++.

Ако са повече от  
една, се използва  
съставен оператор

□ **условие** – логически  
израз определящ до  
кога ще се повтаря  
цикъла

# Действие на командата

1. **do**
  2. *команда;*
  3. **while (условие);**
  4. *следваща команда;*
1. изпълнява се командата
  2. проверява се условието
  3. ако върне стойност **true**:  
цикълът се повтаря отново
  4. ако върне стойност **false**:  
цикълът се прекратява и  
се преминава на  
следващата команда

# Особености

- понеже проверката за край се извършва след изпълнението на командата, то тялото на цикъла ще се изпълни поне веднъж
- в тялото на цикъла трябва да има команди, които променят условието, в противен случай цикъла ще се изпълнява до безкрай

# Съставни части на една програма с цикъл с постусловие:

1. декларации;
2. инициализации;
3. **do**
4. **{**
5.     тяло на цикъла;
6.     актуализация;
7. **}**
8. **while** (условие за край);
9. отпечатване;

Проверка дали е въведена валидна стойност:

1. повтаряме:
2.     въвеждаме стойността;
3.     ако не е валидна – грешка;
4.     докато (стойността не е валидна);

Проверка дали е въведено положително число:

1. `int x;`
2. `do {`
3.     `cout<<"Vavedete polozitelno chislo: ";`
4.     `cin>>x;`
5.     `if (x<=0) cout<<"Greshna stoinost!\n"`
6. `} while (x<=0);`

Съставни части на една програма с цикъл с постусловие:

1. декларации;
2. инициализации;
3. do
4. {
5.     актуализация;
6.     тяло на цикъла;
7. }
8. while (условие);
9. отпечатване;

Въвеждане на числа до въвеждане на 0 и отпечатване на сбора им:

1. int x,sum;
2. sum=0;
3. do
4. {
5.     cin>>x;
6.     sum+=x;
7. }
8. while (x!=0);
9. cout<<sum<<endl;

Съставни части на една програма с цикъл с постусловие:

1. декларации;
2. инициализации;
3. do
4. {
5.     тяло на цикъла;
6.     актуализация;
7. }
8. while (условие);
9. отпечатване;

Намерете **x** на степен **y**, където **x** и **y** са положителни цели числа:

степен           0    1    2    3    y  
резултат=1 . x . x . x ... x

1. int x,y,p;
2. cin>>x>>y;
3. p=1;
4. do
5. {
6.     p=p\*x;
7.     y--;
8. }
9. while (y>0);
10. cout<<p<<endl;



Съставни части на една програма с цикъл с постусловие:

1. декларации;
2. инициализации;
3. do
4. {
5.     тяло на цикъла;
6.     актуализация;
7. }
8. while (условие);
9. отпечатване;

Да се въведе число **x** и цифра **y** и да се изведе “da” ако **x** съдържа **y** и “ne” – ако не я съдържа:

1. int x,y,N;
2. cin>>x>>y;
3. do
4. {
5.     N=x%10;
6.     x/=10;
7. }
8. while ((x!=0)&&(N!=y));
9. if(N==y) cout<<"da\n";
10. else cout<<"ne\n";

# Задачи

1. Да се задава въпрос докато на него не се отговори с Y или N
2. Да се отпечата числата от 1 до N и сумата им
3. Да се отпечата всички четни положителни числа по-малки от N
4. Да се отпечата всички степени на x по-малки от y
5. Да се позволи въвеждане на числа до въвеждане на 0 и да се отпечата произведението им
6. Да се позволи въвеждане на числа до въвеждане на 0 и да се отпечата броят им
7. Да се позволи въвеждане на числа до въвеждане на 0 и да се отпечата най-малкото и най-голямото от тях

# Задачи

8. Да се позволи въвеждане на числа докато не се въведат 3 положителни.
9. Да се въведе число и да се отпечатаат цифрите му в обратен ред
10. Да се въведе число и да се отпечата сумата от цифрите му и колко цифрено е
11. Да се пресметне сумата между число и огледалното му число
12. Да се позволи въвеждане на число и цифра X и да се отпечата дали цифрата се съдържа в записа на числото
13. Да се позволи въвеждане на числа до въвеждане на 0 и да се отпечата броят на четните, положителните, едноцифрените, двуцифрените и кратните на 3 и 5

# Задачи

14. Да се позволи въвеждане на числа до въвеждане на 0 и да се отпечата имало ли е въвеждане на поредни еднакви числа и ако да – кое число и колко пъти е повторено
15. Да се въведе число и да се отпечата DA ако всичките му цифри са в нарастващ ред (например 1234 или 24689)
16. Да се въведе число и да се отпечата, разложено на прости множители или текста "prosto chislo" ако няма такива (например  $24=2*2*2*3$  или  $17=\text{prosto chislo}$ )
17. Да се позволи въвеждане на число в десетична бройна система и да се преобразува в двоична
18. Да се позволи въвеждане на число в двоична бройна система и да се преобразува в десетична (числото ще си е десетично, но ще е записано само с 0 и 1 и ще го третираме като двоично, например  $1101=13$ )

# Задачи

19. Имаме числова ос с числа от 1 до 100. Да се позволи въвеждане на двойки числа, указващи началото и края на отсечки от тази ос, докато не се въведат 0 0. Да се отпечатаат координатите на най-късата и най-дългата, най-лявата и най-дясната отсечки, както и на тази, която обхваща всички други (ако има такава)
20. Редицата 1, 1, 2, 3, 5, 8, 13, 21... при която всяко число е сума на предните две, се нарича редица на Фибоначи. Да се напише програма, която при въвеждане на  $n$  намира:
  - $n$ -тото число на Фибоначи (например при  $n=8$  извежда 21)
  - най-близкото число от редицата на Фибоначи, което е по-голямо от  $n$  (например при  $n=15$  извежда 21)

# И В ЖИВОТА:

---

```
do {  
    опитай;  
    направи си изводи;  
} while (не стане);
```