

Вложени цикли



Определение

- Когато в тялото на един оператор за цикъл е вложен друг оператор за цикъл, говорим за **вложени цикли**. Единият се нарича **външен**, а другият – **вътрешен**.

Действие

- За всяко завъртане на външния цикъл вътрешния се извърта целия.

Пример за вложени цикли

Таблицата за умножение до 10:

```
for(int i=1;i<=10;i++)  
    for(int j=1;j<=10;j++)  
        cout<<i<<`*`<<j<<`= `<<i*j<<endl;
```

Правила при вложени цикли

- Управляващите променливи на двата цикъла трябва да са различни
- Вътрешният цикъл трябва да се съдържа изцяло в тялото на външния
- От тялото на вътрешния цикъл можем да направим преход към тялото на външния, но не и обратно
- Вътрешният и външният цикъл може да са от един и същи тип, или от различни типове
- Променливи, декларирани в тялото на вътрешния цикъл, не са видими извън него

Да се въведе **n** и да изведе на екрана

СЛЕДНОТО:

1	int n; // декларираме променливите
1 2	cin>>n; // въвеждаме числото
1 2 3	for(int i=1;i<=n;i++) // цикъл за редовете
.....	{ // за всеки ред ще изпълнява няколко неща
.....	for(int j=1;j<=i;j++) // цикъл за колоните
1 2 3 n	cout<<j; // отпечатваме цифрата
	cout<<endl; // нов ред накрая на реда
	} // това се повтаря за всеки ред

Да се въведе **n** и да изведе на екрана

СЛЕДНОТО:

1

2 3

3 4 5

4 5 6 7

.....

n n+1 .. 2*n-1

```
int n;           // декларираме променливите
cin>>n;         // въвеждаме числото
for(int i=1;i<=n;i++) // цикъл за редовете
{ // за всеки ред ще изпълнява няколко неща
  for(int j=i;j<2*i;j++) // цикъл за колоните
    cout<<j; // отпечатваме цифрата
  cout<<endl; // нов ред накрая на реда
} // това се повтаря за всеки ред
```

Въвеждане на числа до нула и отпечатване на броят на простите числа между тях

```
□ int chislo, br=0;           // декларираме променливите
□ bool prosto;              // флаг дали числото е просто
□ do {                      // повтаряме:
□   cin >> chislo;          // въвеждаме числото
□   prosto=true;           // предполагаме че е просто
□   for(int i=2; i<=chislo/2; i++) // проверяваме дали е така
□     if(chislo%i==0) {     // ако се дели на друго...
□       prosto=false;      // ...значи не е просто
□       break;            // прекъсваме проверката
□     }
□   if (prosto)&&(chislo!=0) br++; // броим само простите
□ } while (chislo!=0);      // докато не въведем 0
□ cout << br << endl;      // отпечатваме го
```

Въвеждане на k и n числа и намиране на k-цифрените между тях

```
□ int chislo, k, n, br;      // декларираме променливите
□ cin >> k >> n;           // въвеждаме входните данни
□ for(int i=1; i<=n; i++) { // повтаряме n пъти
□   cin >> chislo;          // въвеждаме числото
□   br=0;                  // инициализация на цикъла
□   while (chislo!=0) {    // докато не свърши числото
□     chislo/=10;          // изтриваме последната цифра
□     br++;                // и я отброяваме
□   }
□   if (br==k)              // ако е с k цифри...
□     cout << "tova e!" << endl; // ... отпечатваме че е познато
□ }
```


Извода:



от толкова цикли вече
зациклихме