

## В информатиката всичко реално се свежда до преобразуване на данните!

В тези задачи ще срещнете основни линейни логически преобразувания

За да образуваме необходимите числа най-често можем да използваме един от следните 3 похвата:

- 1) Пряка зависимост на индекса с числото, което искаме да получим
- 2) Допълнителна помощна променлива
- 3) Използване на зависимост спрямо предходни елементи

За самото обхождане на елементите търсим някакво повторение.

- a) Най-простото повторение би било ако обхождам елементите един по един задавайки директно стойностите. Въртим цикъла директно по индексите и в цикъла направо задаваме стойността която желаем
- b) По-сложно повторение би било ако имаме примерно редуващи се последователности. Примерно 0,1,2,3,4, 0,1,2,3,4, 0,1,2,3,4 или пък 3,7, 3,7, 3,7 и т.н. В такъв случай можем да въртим цикъл със стъпка броя на елементите и да ги задаваме директно като поредица в единично завъртане (една итерация).

### Решения на задачи в няколко различни варианта:

#### 1 Задача

Да се образува масив от 10 елемента, които са: **20 25 30 35 40 45 50 55 60 65**

#### 2 Задача \*

Да се образува масив от 10 елемента, които са: **2 3 5 8 13 21 34 55 89 144**

(всеки един е сумата от предходните два)

#### 3 Задача

Да се образува масив от 20 елемента, които са: **3 7 3 7 3 7 3 7 ...**

#### 4 Задача

Да се образува масив от 20 елемента, които са: **0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 ...**

#### 5 Задача

Да се образува масив от 20 елемента, които са: **5 -6 7 -8 9 -10 11 -12 и т.н.**

#### 6 Задача

Да се образува масив от 21 елемента, които са: **10 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10**

#### 7 Задача

Да се образува масив от първите 50 елемента, които се делят или на 3 или на 5 или и на двете числа  
**0 3 5 6 9 10 12 15 .... и т.н.**

#### 8 Задача

Да се образува масив от първите 30 елемента, така че да са: **-5 5 -7 7 -9 9 -11 11 ... и т.н.**

#### 9 Задача

Да се образува масив от първите 30 елемента, така че да са: **7 7 7 8 8 8 9 9 9 и т.н.**

\* Навсякъде съм разделил числата с интервали, а не запетайки за да се събират на един ред

## 1 Задача Да се образува масив от 10 елемента, които са: 20 25 30 35 40 45 50 55 60 65

```
int[] myArray = new int[10]; //задължително декларираме празен масив
//тази част би се различавала евентуално само по броя елементи
//и името, което трябва да зададете на масива

//Задача 1 => Вариант 1а: чрез директно задаване на елементите през зависимост
for (int x = 0; x < 10; x++)
{
    myArray[x] = 5 * x + 20; //или 5*(x+4)
}

//Задача 1 => Вариант 2а: чрез директен обход на елементите + помощна променлива
for (int x = 0, a = 20; x < 10; x++, a += 5)
{
    myArray[x] = a;
}

//Задача 1 => Вариант 3а: чрез директен обход и зависимост от предходен елемент
myArray[0] = 20;
for (int x = 1; x < 10; x++) //Важно! Тук не въртим от 0 а от 1 иначе x-1 в тялото ще гръмне
{
    myArray[x] = myArray[x - 1] + 5;
    //нека текущия елемент да бъде с 5 по-голям от предходния
}
```

## 2 Задача\* Да се образува масив от 10 елемента, които са: 2 3 5 8 13 21 34 55 89 144

Тук, ако се загледаме по-внимателно виждаме, че всеки седмач елемент реално е СБОР от предходните два. Как трябва да постъпим? Ще използваме трети похват – зависимост от предходните елементи.

```
//Задача 2 => Похват 3а: чрез директен обход и базирани на предходни елементи
int[] myArray = new int[10];

myArray[0] = 2;
myArray[1] = 3;
for (int x = 2; x < 10; x++) //Тук въртим директно от 2 иначе x-2 в тялото ще гръмне
{
    myArray[x] = myArray[x-2] + myArray[x-1];
    //нека текущия елемент да бъде сумата на предходните 2
}
```

### 3 Задача

Да се образува масив от 20 елемента, които са: 3 7 3 7 3 7 ...

Тази ще я решим по 3 начина:

```
//Задача 3 => Похват 1a: директен обход, като стойността е функция/зависимост от индекса
int[] myArray = new int[20];

for (int x = 0; x < 20; x++)
{
    if (x%2==0) myArray[x] = 3;
    else myArray[x] = 7;
}

//Задача 3 => Похват 2б: обход през стъпка равна на повторението и директно задаване
for (int x = 0; x <= 18; x+=2)
{
    myArray[x] = 3;
    myArray[x+1] = 7;
}

//Задача 3 => Похват 3а: обход по елементи и задаване чрез образуван вече предходен елемент
myArray[0] = 3;
for (int x = 1; x <= 19; x++) // <=19 тук е аналогично на <20
{
    myArray[x] = 10 - myArray[x-1]; //сбора на два съседни елемента винаги е ДЕСЕТ
}
```

### 4 Задача

Да се образува масив от 20 елемента, които са: 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 ...

```
//Задача 4 => Похват 1a: директен обход и задаване на стойностите
int[] myArray = new int[20];

for (int x=0; x<20; x++)
{
    myArray[x] = x % 5;
}

//Задача 4 => Похват 2б: обход със стъпка спрямо повторението и задаване по няколко наведнъж
for (int x=0; x<=15; x += 5)
{
    myArray[x] = 0;
    myArray[x+1] = 1;
    myArray[x+2] = 2;
    myArray[x+3] = 3;
    myArray[x+4] = 4; //когато x=15 този ще зададе 19-тия т.е. последния елемент
}

//Задача 4 => Похват 3а: директно обхождане, чрез използване на предходните
myArray[0] = 0;
for (int x = 1; x <= 19; x++)
{
    myArray[x] = myArray[x-1] + 1;
    if (myArray[x] == 5) myArray[x] = 0; //ако стане 5 да го върнем на нула
}
```

5 Задача Да се образува масив от 20 елемента, които са: 5 -6 7 -8 9 -10 11 -12 и т.н.

```
//Задача 5 => Похват 2а: Обхождаме по индекс и използваме допълнителна променлива
for (int x = 0; x < 20; x++)
{
    int a = x+5; //създава всеки път нова променлива
    if (x%2 == 0) myArray[x] = -a;
    else myArray[x] = a;
}
```

6 Задача Да се образува масив от 21 елемента, които са: 10 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10

```
//Задача 6 => (21 елемента) 10 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10
//Решение 1 => Обхождаме индексите чрез 2 отделни цикъла
for (int x = 0; x < 11; x++)
{
    myArray[x] = 10-x; //при малките индекси числата намаляват
}

for (int x = 11; x < 21; x++)
{
    myArray[x] = x-10; //при големите индекси се увеличават
}

//Решение 2 => Обхождаме индексите направо в един цикъл
for (int x = 0; x < 21; x++)
{
    if (x<11) myArray[x] = 10-x;
    else myArray[x] = x-10;
}
```

7 Задача \*\*

Да се образува масив от първите 50 елемента, които се делят или на 3 или на 5 или и на двете числа 0 3 5 6 9 10 12 15 ... и т.н.

```
//Задача 7**:
int c = 0; //хем ги брои хем показва следващата празна позиция в масива
int v = 0; //обхождаме числата нагоре и сравняваме дали се делят на 3 или 5
while (c < 50)
{
    if (v % 3 == 0 || v % 5 == 0)
    {
        myArray[c] = v;
        c++; //увеличаваме само когато добавим ново число
    }
    v++; //увеличаваме всеки път търсейки следващото
}

//Задача 7** Вариант 2 - със Списък
List<int> values = new List<int>();
int z = 0; //за обход на числата нагоре
while (values.Count < 50) //докато не сме добавили толкова елементи в списъка
{
    if (z%3==0 || z%5==0) values.Add(z);
    z++;
}

myArray = values.ToArray();
```

## 8 Задача

Да се образува масив от първите 30 елемента, така че да са: **-5 5 -8 8 -11 11 -14 14...** и т.н.

```
//Задача 8 => Похват 2б – и с помощна променлива, и с обход през стъпка
for (int x=0, a=5; x<=28; x+=2, a+=3)
{
    //стъпката за индексите е през 2, а на помощната променлива през 3
    myArray[x] = a;
    myArray[x+1] = -a;
}

//Обърнете внимание, че тук не можем да декларираме променливата "a" вътре
//в тялото на цикъла подобно на Задача 5,
//понеже тук a и x нямат такава директна връзка както там a = x+5
//* реално пак имат, но е доста неинтуитивна a = 3*(x/2)+5
```

## 9 Задача

Да се образува масив от първите 30 елемента, така че да са: **7 7 7 8 8 8 9 9 9** и т.н.

```
//Задача 9 => Похват 2б: С помощна променлива + образуване по групи

int a = 7; //помощната променлива може и в масива да се декларира
for (int x=0; x<30; x += 3) //последното x ще е 27 и ще образува 27,28,29
{
    myArray[x] = a;
    myArray[x+1] = a;
    myArray[x+2] = a;
    a++;
}

//Задача 9 => Един по-трики метод за Похват 2а (само с помощна променлива)
for (int x=0; x<30; x++)
{
    int num = x/3;
    //понеже и двете са цели числа резултата е отрязан до цялото число
    myArray[x] = num + 7; //добавяме изместването от 7 единици
}
}
```