



# Упражнение

доц. д-р Нора Ангелова

---

## Какво ще изведе

```
int i = 1;
```

```
int a = ++i + 2;
```

```
cout << a;
```

Резултат: 4

## Какво ще изведе

```
int i = 1;
```

```
int a = i++ + 2;
```

```
cout << a;
```

Резултат: 3

```
cout << i;
```

Резултат: 2

## КАКВО ЩЕ ИЗВЕДЕ

```
int f(int& x) {  
    x++;  
    return x;  
}
```

```
int main() {  
    int a = 5;  
    cout << f(a) << endl;  
    return 0;  
}
```

**6**

## КАКВО ЩЕ ИЗВЕДЕ

```
int f(int& x) {  
    return ++x;  
}
```

```
int main() {  
    int a = 5;  
    cout << f(a) << endl;  
    return 0;  
}
```

**6**

## Какво ще изведе

```
int f(int& x) {  
    return ++x;  
}
```

```
int main() {  
    int a = 5;  
    cout << f(a+1) << endl;  
    return 0;  
}
```

**Error:** cannot convert  
parameter 1 from  
'int' to 'int &'

## КАКВО ЩЕ ИЗВЕДЕ

```
int f(int& x) {  
    return x++;  
}
```

```
int main() {  
    int a = 5;  
    cout << f(a) << endl;  
    cout << a << endl;  
  
    return 0;  
}
```

**5**  
**6**

## КАКВО ЩЕ ИЗВЕДЕ

```
int f(const int& x) {  
    return x;  
}  
  
int main() {  
    int a = 5;  
    cout << f(a+1) << endl;  
    return 0;  
}
```

**6**



## Какво ще изведе

```
char str[10];  
char str2[10];  
  
cin.get(str, 10, ' ');  
cin.get(str2, 10, ' ');  
  
cout << str << endl;  
cout << str2 << endl;
```

**ВХОД: 1234 567**

**1234**

**празно**

## Какво ще изведе

```
char str[10];  
char str2[10];  
  
cin.getline(str, 10, ' ');  
cin.get(str2, 10, ' ');  
  
cout << str << endl;  
cout << str2 << endl;
```

**ВХОД: 1234 567**

**1234**

**567**

## Какво ще изведе

```
char str[10];  
char str2[10];  
  
cin.get(str, 10, ' ');  
  
char s1;  
cin >> s1;  
  
cin.get(str2, 10, ' ');  
  
cout << str << endl;  
cout << str2 << endl;
```

**ВХОД: 1234 567**

**1234**

**67**

## Какво ще изведе

```
char str[10];  
char str2[10];  
  
cin.get(str, 10, ' ');  
  
char s1;  
cin.get(s1);  
  
cin.get(str2, 10, ' ');  
  
cout << str << endl;  
cout << str2 << endl;
```

**ВХОД: 1234 567**

**1234**

**567**

# Рекурсия

## Задача

Дадено е неотрицателно цяло число. Да се дефинира рекурсивна функция, която намира броя на цифрите на числото в бройна система с основа  $k$ .

```
int countDig(unsigned number, int k) {  
    if (number < k) {  
        return 1;  
    }  
  
    return countDig(number/k, k) + 1;  
}
```

# Рекурсия

## Задача

Да се дефинира рекурсивна функция, която заменя всяко срещане на цифрата 5 в дадено неотрицателно цяло число с 8.

```
int replaceFive(unsigned number) {
    if (number == 0) {
        return 0;
    }
    // Може да използвате тернарен оператор
    int lastDigit = number % 10;
    if (lastDigit == 5) {
        lastDigit = 8;
    }
    return replaceFive(number/10)*10 + lastDigit;
}
```

# Рекурсия

## Задача

Лабиринт е представен с булева квадратна матрица  $8 \times 8$ . Клетка се приема за проходима, ако елементът в съответната позиция е истина и за непроходима в противен случай.

Да се напише програма, която проверява дали съществува път от съседни в хоризонтално и вертикално направление проходими клетки на лабиринта, който започва в горния му ляв ъгъл и завършва в долния му десен ъгъл.

# Рекурсия

```
bool labyrinth[8][8] = {
    {1, 0, 0, 0, 1, 1, 1, 1},
    {1, 0, 0, 0, 1, 1, 1, 1},
    ...
    {1, 0, 0, 0, 1, 1, 1, 1},
};
// ...
bool way(bool labyrinth[][8], int size, int x, int y) {
    // напуснали сме границите на лабиринта
    if (x < 0 || y < 0 || x > size-1 || y > size-1) {
        return false;
    }

    // целта е достигната
    if (x == size-1 && y == size-1) {
        return true;
    }

    // клетката е непроходима
    if (!labyrinth[x][y]) {
        return false;
    }

    // обявяваме клетката за обходена, за да се предотврати зацикляне
    // Ще го разгледаме на следващата лекция
    labyrinth[x][y] = 0;

    //търсене на път от някой от четирите съседа
    return way(labyrinth, size, x+1, y) ||
           way(labyrinth, size, x, y+1) ||
           way(labyrinth, size, x-1, y) ||
           way(labyrinth, size, x, y-1);
}
```



```
cout << “Край”;
```