

Использование локальных и глобальных переменных

Переменные, объявляемые внутри функции, называются **локальными** переменными. Их значения и даже сам факт, что локальные переменные существуют, известны только данной функции. Имена переменных, используемых внутри функций, должны быть уникальными только по отношению к данной функции.

Глобальные переменные в отличие от локальных переменных известны на протяжении всей программы и доступны внутри всех функций. Глобальные переменные чрезвычайно полезны, когда одни и те же данные используются в нескольких функциях программы.

Глобальные переменные необходимо использовать осторожно:

- они используют память в течение всего времени работы программы, а не тогда, когда они необходимы;
- использование глобальных переменных вместо локальных приводит к тому, что функции становятся более частными, поскольку они зависят от переменных, определяемых снаружи;
- использование большого числа глобальных переменных может вызвать ошибки в программе из-за неизвестных и нежелательных эффектов.

Задача 1

Программа формирует случайным образом два целочисленных массива, состоящих из 12 элементов в диапазоне от 0 до 100.

По числу, введенному с клавиатуры, определяет и выводит на экран количество элементов, больших этого числа.

1.	<pre>#include <iostream> #include <cstdlib> #include <ctime> #include <cmath> using namespace std;</pre>	
2.	<pre>int KZ (int fC[12], int fY) { int i, k=0; for (i=0; i<12;i++) if (fC[i]>fY) k++; return k; }</pre>	Функция KZ определяет количество элементов массива fA, больших числа fZ.
3.	<pre>int main() { srand (time (NULL)); int i, Z, Rez; int A[12], B[12];</pre>	Начало программы. Описание переменных.
4.	<pre>cout<<"\n\nA:\n\n"; for (i=0; i<12;i++) {A[i] = rand ()%101; cout<< A[i] <<"\t";}</pre>	Формирование и печать массива A.
5.	<pre>cout<<"\n\nB:\n\n"; for (i=0; i<12;i++) {B[i] = rand ()%101; cout<< B[i] <<"\t";}</pre>	Формирование и печать массива B.
6.	<pre>cout<<"\n\nZ=? "; cin >>Z;</pre>	Ввод с клавиатуры числа-ограничителя.
7.	<pre>Rez=KZ(A,Z); cout<<"\n Kol-vo A[i]>" <<Z<< " = " <<Rez;</pre>	Обращение к функции для массива A.

8.	Rez=KZ(B,Z); cout<<"\n Kol-vo B[i]>" <<Z<<" = "<<Rez;	Обращение к функции для массива B.
9.	return 0; }	Конец

```

A:
43      61      32      60      41      37      32      63      13      76
98      76

B:
88      41      16      45      29      55      32      50      96      31
58      92

Z=? 50

Kol-vo A[i]>50 = 6
Kol-vo B[i]>50 = 5
Process returned 0 (0x0)   execution time : 11.797 s
Press any key to continue.

```

Задача 2

Рассмотрим ту же задачу, но будем использовать глобальную переменную Z.

1.	#include <iostream> #include <cstdlib> #include <ctime> #include <cmath> using namespace std;	
2.	int Z;	Объявление глобальной переменной
3.	int KZ (int fC[12]) { int i, k=0; for (i=0; i<12;i++) if (fC[i]>Z) k++; return k; }	Функция KZ определяет количество элементов массива fA, больших числа Z.
4.	int main() { srand (time (NULL)); int i, Rez; int A[12], B[12];	Начало программы. Описание переменных.
5.	cout<<"\n\nA:\n\n"; for (i=0; i<12;i++) {A[i] = rand ()%101; cout<< A[i] <<"\t";}	Формирование и печать массива A.
6.	cout<<"\n\nB:\n\n"; for (i=0; i<12;i++) {B[i] = rand ()%101; cout<< B[i] <<"\t";}	Формирование и печать массива B.
7.	cout<<"\n\nZ=? "; cin >>Z;	Ввод с клавиатуры числа-ограничителя.

8.	Rez=KZ(A); cout<<"\n Kol-vo A[i]>" <<Z<<" = "<<Rez;	Обращение к функции для массива А.
9.	Rez=KZ(B); cout<<"\n Kol-vo B[i]>" <<Z<<" = "<<Rez;	Обращение к функции для массива В.
10.	return 0; }	Конец