

Задача F. Цифры

На вход программе подается последовательность чисел от 1 до 9, заканчивающаяся нулем.
Всего будет введено не более 100000 чисел.
Подсчитайте в этой последовательности количество единиц, количество двоек, количество троек и т.д. и выдайте результат.
В выходных данных всегда должно быть 9 чисел.

Примеры
входные данные
1 1 4 1 5 8 6 3 5 1 0

выходные данные
4 0 1 1 2 1 0 1 0



внутри функций максимум 2мб = 2000 000

байтов = 500 000 int

глобально максимум 1.7гб = 400 000 000 int

16ГБ и винда

сервер 500 ГБ оперативки с Linux 80000 p

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <ctime>

using namespace std;
void Massiv_Generator(int ms[], int razmer)
{
    int i=0;
    while(i<razmer)
    {
        ms[i]=rand()%9;
        i=i+1;
    }
}
void printmass(int ms[], int razmer)
{
    int i=0;
    while(i<razmer)
    {
        printf("%d ", ms[i]);
        i=i+1;
    }
    printf("\n");
}
int Repeated_elements(int ms[], int razmer, int param)
{
    int result1=0;
    int i=0;
    while(i<razmer)
    {
        if(ms[i]==param)
        {
            result1=result1+1;
        }

        i=i+1;
    }
    return result1;
}
int main()
{
    srand(time(NULL));
    int ms[20]={};
    Massiv_Generator(ms, 20);
    printmass(ms, 20);
    int s1=0;
    s1=Repeated_elements(ms, 20, 1);
    int s2=0;
    s2=Repeated_elements(ms, 20, 2);
    int s3=0;
    s3=Repeated_elements(ms, 20, 3);
    int s4=0;
    s4=Repeated_elements(ms, 20, 4);
    int s5=0;
    s5=Repeated_elements(ms, 20, 5);
    int s6=0;
    s6=Repeated_elements(ms, 20, 6);
    int s7=0;
    s7=Repeated_elements(ms, 20, 7);
    int s8=0;
    s8=Repeated_elements(ms, 20, 8);
    int s9=0;
    s9=Repeated_elements(ms, 20, 9);
    printf("%d %d %d %d %d %d %d %d %d\n", s1, s2, s3, s4, s5, s6, s7, s8, s9);
}
```

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <ctime>
#define SIZE 5000

using namespace std;
void Massiv_Generator(int ms[], int razmer)
{
    int i=0;
    while(i<razmer)
    {
        ms[i]=rand()%9+1;
        i=i+1;
    }
}
void printmass(int ms[], int razmer)
{
    int i=0;
    while(i<razmer)
    {
        printf("%d ", ms[i]);
        i=i+1;
    }
    printf("\n");
}
void Repeated_elements(int ms[], int razmer, int msresult[])
{
    int i=0;
    while(i<razmer)
    {
        msresult[ms[i]-1]+=1;
        i=i+1;
    }
}
int main()
{
    srand(time(NULL));
    int msresult[9]={};
    int ms[SIZE]={};
    Massiv_Generator(ms, SIZE);
    printmass(ms, SIZE);
    Repeated_elements(ms, SIZE, msresult);
    printmass(msresult, 9);
}
```

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <ctime>
#define SIZE 400000000

using namespace std;
void Massiv_Generator(int ms[], int razmer)
{
    int i=0;
    while(i<razmer)
    {
        ms[i]=rand()%9+1;
        i=i+1;
    }
}
void printmass(int ms[], int razmer)
{
    int i=0;
    while(i<razmer)
    {
        printf("%d ", ms[i]);
        i=i+1;
    }
    printf("\n");
}
void Repeated_elements(int ms[], int razmer, int msresult[])
{
    int i=0;
    while(i<razmer)
    {
        msresult[ms[i]-1]+=1;
        i=i+1;
    }
}
int ms[SIZE]={};
int main()
{
    srand(time(NULL));
    int msresult[9]={};
    Massiv_Generator(ms, SIZE);
    //printmass(ms, SIZE);
    Repeated_elements(ms, SIZE, msresult);
    printmass(msresult, 9);
}
```