Богатов Р.Н.

Программирование на языке высокого уровня

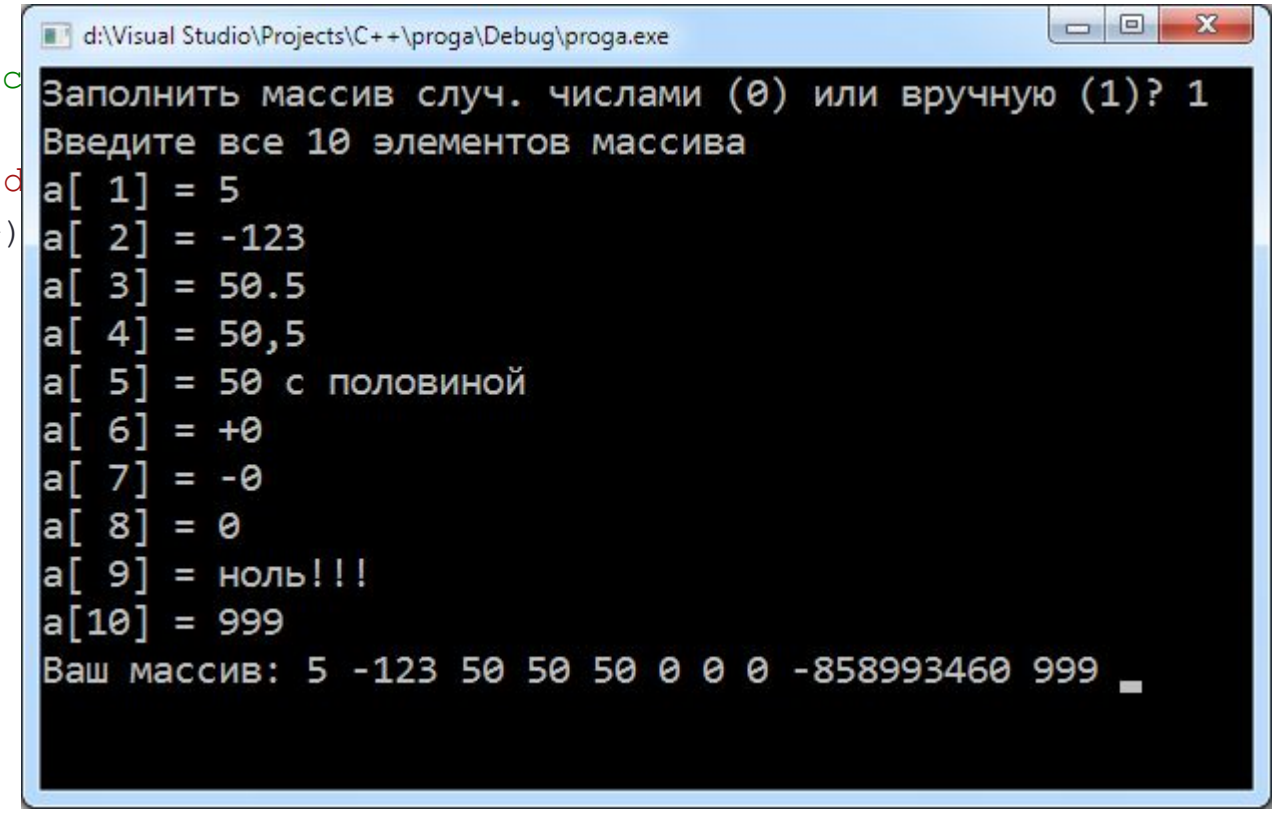
C++ ► Лекция 4.5 ► Примеры задач, которые решаются с помощью массивов

Кафедра АСОИУ ОмГТУ, 2016

Инициализация массива

```
...
if (mode==0) // случайное заполнение массива
{
  srand( (unsigned)time(NULL) );
  for(int i=0; i<N; i++)
    a[i] = rand();
}
else // ручной ввод масс
{
  printf("Введите все %d
  for(int i=0; i<N; i++)
    printf("a[%2d] = ",
    scanf("%d", &a[i]);
    fflush(stdin);
  }
}

// вывод массива
printf("Ваш массив: ");
for(int i=0; i<N; i++)
  printf("%d ", a[i]);
...
```



Задача про пятаки и трёшки

```
// прямое решение на основе остатка от деления
int A, B; // искомое число пятаков и трёшек

for (int i = 0; i < 5; i++)
    if ((n - i * 3) % 5 == 0)
        B = i;

A = (n - B * 3) / 5;
```

```
int A, B; // искомое число пятаков и трёшек

for (B = 0; ; B++)
    if ((n - B * 3) % 5 == 0)
        break;

A = (n - B * 3) / 5;
```

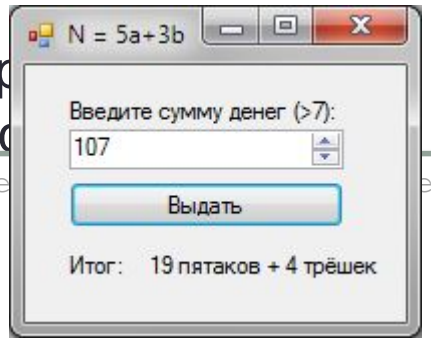
=

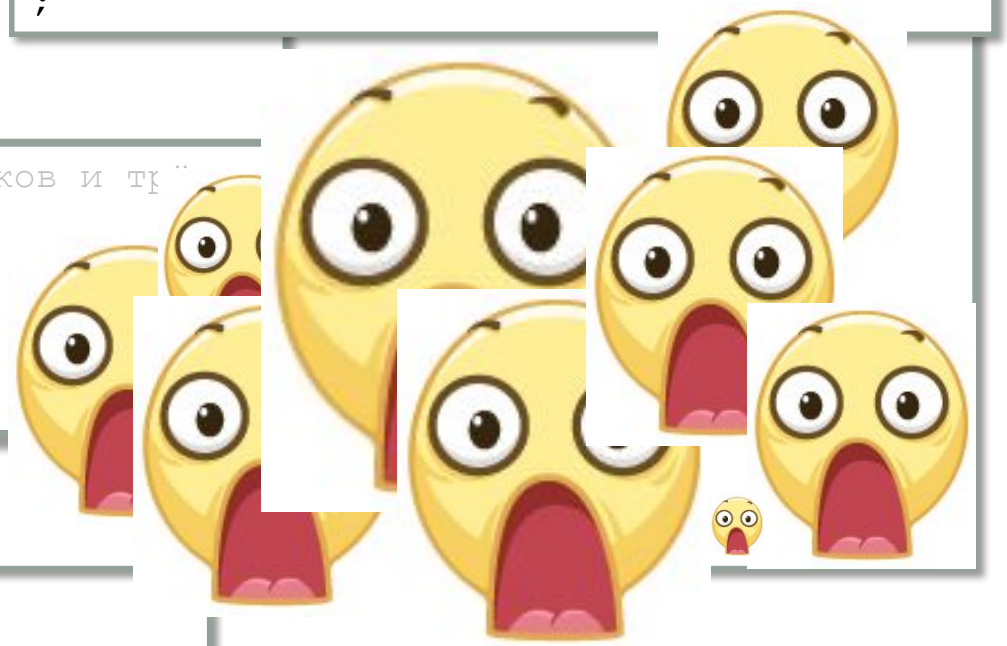
```
for (B = 0; (n - B * 3) % 5 != 0; B++)
;
```

```
int A, B; // искомое число пятаков и тр

int G[5] = { 0, 2, 4, 1, 3 };
B = G[n % 5];
A = (n - B * 3) / 5;
```

```
else
{
    A = (n - 9) / 5;
    B = 3;
}
```





Произведение матриц

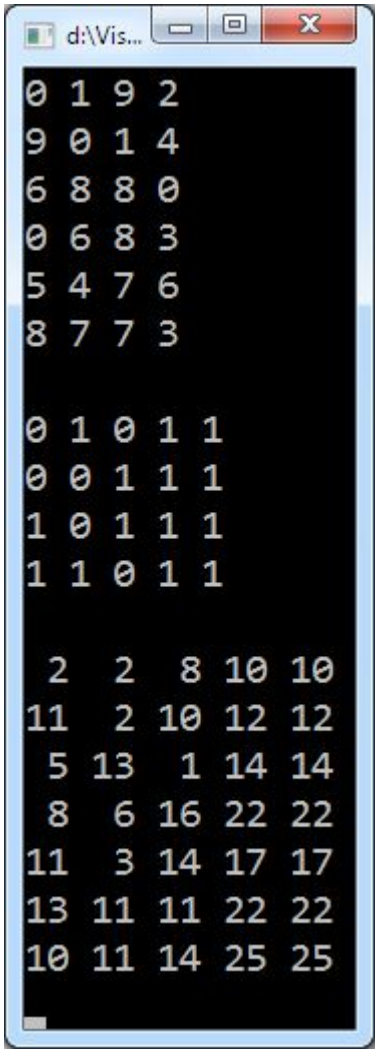
```
// произведение прямоугольных матриц константного размера
const int N=7, M=4, L=5;
int a[N][M], b[M][L], c[N][L];

// инициализация массива a
...
// вывод массива a на экран
...
// инициализация массива b
...
// вывод массива b на экран
...

// произведение матриц
for(int i=0; i<N; i++)
  for(int j=0; j<L; j++)
  {
    int s = 0;
    for(int k=0; k<M; k++)
      s += a[i][k]*b[k][j];

    c[i][j] = s;
  }

// вывод результата на экран
...
```



Перебор всех расстановок

Математическая постановка задачи: по
расстановок единиц в множестве из *N*

Постановка задачи для программиста:
заполнения массива длины *N* нулями и

```
// частный случай: получение всех расстановок еди
// в массиве длиной N<31

  ...
  for(int n=0; n<(1<<N); n++)
  {
    for(int i=0; i<N; i++)
      a[i] = (n & (1<<i))>0 ? 1 : 0;

    // вывод массива
    printf("\r%d: ", n);
    for(int i=0; i<N; i++)
```

Вопросы:

1) Почему алгоритм для общего случая работает быстрее?

2) Почему во втором алгоритме массив нужно выводить до обработки массива?

3) Какой из алгоритмов удобнее взять за основу для решения задачи расстановки трёхзначных чисел?

```
// общий случай: получение всех расстановок
// единиц в массиве произвольной длины

#include <stdio.h>
#include <conio.h>

const int N=10;
int a[N];

void main()
{
  for(int n=0; n<(1<<N); n++)
  {
    // вывод массива
    printf("\r%d: ", n);
    for(int i=0; i<N; i++)
      printf("%d", a[i]);

    for(int i=0; i<N; i++)
      if (a[i]==1) // переполнение
        a[i] = 0;
      else // прибавление единицы
      {
        a[i] = 1;
        break;
      }
  }
  _getch();
}
```