

```
#include <stdio.h>
#include <string.h>

int sort(const
{
    ret
}

int ma
{
    int i, a =
    char **A;
    char b[1025];
    FILE *input, *out;
```

По инициативе и поддержке ФАКТ МФТИ

# Курс молодого бойца

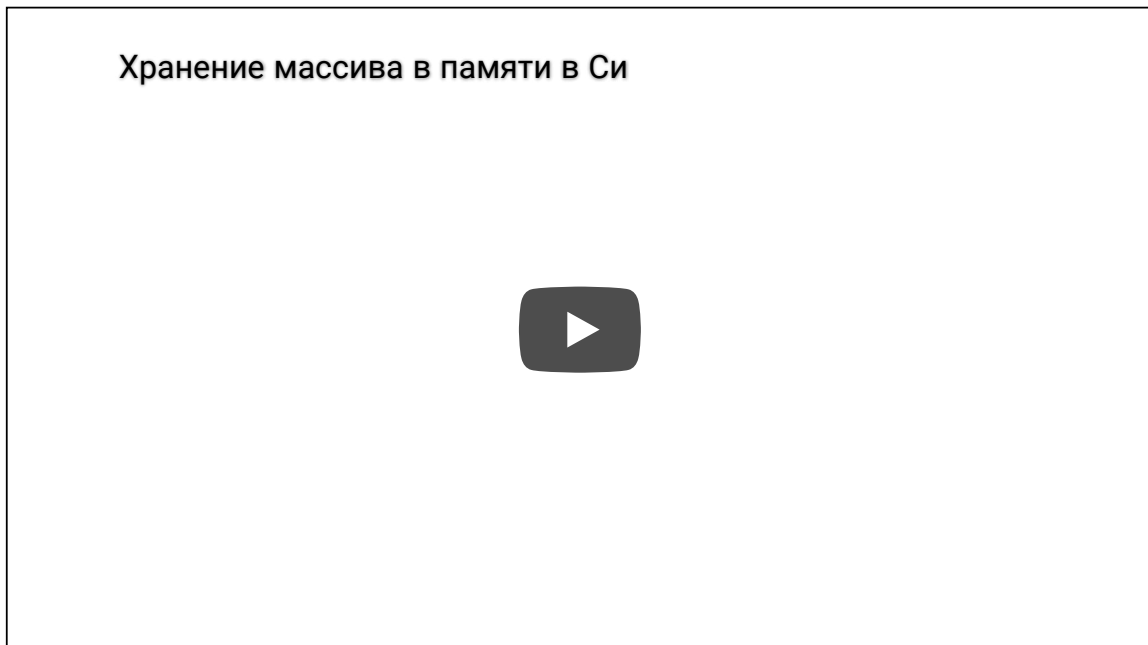
Подготовка к информатике на 1-м курсе (язык Си)

Хирьянов Тимофей Федорович  
Кафедра информатики и вычислительной математики МФТИ

[Главная](#)

# Массивы чисел

Хранение массива в памяти



Что такое массив? Фиксированный размер и однотипность элементов. Хранение в памяти и скорость доступа по индексу.

Создание и заполнение массива

## Создание и заполнение массива на Си



Объявление одномерного массива целых чисел. Заполнение индексами и реверсивными индексами. Специфические заполнения

### array\_init.c

```
#include <stdio.h>

#define N 10

void print_array(int A[])
{
    for(int i = 0; i < N; ++i)
        printf(" %d ", A[i]);
    printf("\n");
}

int main(int argc, char* argv[])
{
    int A[N] = {0};

    for(int i = 0; i < N; ++i) // Task #1
        A[i] = i;
    print_array(A);

    for(int i = 0; i < N; ++i) // Task #2-a
        A[i] = N - 1 - i;
    print_array(A);

    for(int i = 0; i < N; ++i) // Task #2-b
        A[N - 1 - i] = i;
    print_array(A);

    for(int i = 0; i < N; ++i) // Task #3
        A[i] = i%2;
    print_array(A);

    for(int i = 0; i < N/2; ++i) // Task #2-b
    {
        A[2*i] = i;
        A[2*i+1] = N/2 + i;
    }
    print_array(A);

    return 0;
}
```

# Решето Эратосфена

## Решето Эратосфена на Си



Постановка задачи. Оформление решения на Си.

### eratosthenes\_sieve.c

```
#include <stdio.h>
#define N 25

int main(int argc, char* argv[])
{
    int sieve[N] = {0};

    for(int i = 2; i*i < N; ++i)
        if (sieve[i] == 0)
            for(int k = i*i; k < N; k += i)
                sieve[k] = 1;

    for(int i = 0; i < N; ++i)
        printf("%3d", i);
    printf("\n");
    for(int i = 0; i < N; ++i)
        printf("%3d", sieve[i]);
    printf("\n");

    printf("Prime numbers:\n");
    for(int i = 2; i < N; ++i)
        if (sieve[i] == 0)
            printf("%3d", i);
    printf("\n");

    return 0;
}
```

## Копирование массива, реверс и циклический сдвиг

## Копирование массива, реверс циклический сдвиг на Си



Позлементное копирование массива. Реверс массива. Циклический сдвиг влево и вправо в массиве.

### array\_copy.c

```
#include <stdio.h>
#define N 10

void print_array(int A[])
{
    for(int i = 0; i < N; ++i)
        printf("%3d ", A[i]);
    printf("\n");
}

int main(int argc, char* argv[])
{
    int A[N] = {0, 10, 20, 30, 40, 50, 60, 70, 80, 90};
    int B[N] = {0};

    for(int i = 0; i < N; ++i)
        B[i] = A[i];
    print_array(A);
    print_array(B);
    printf("\n");

    for(int i = 0; i < N; ++i)
        B[i] = A[N-1-i];
    print_array(A);
    print_array(B);

    return 0;
}
```

### array\_reverse\_cycle.c

```
#include <stdio.h>
#define N 10

void print_array(int A[])
{
    for(int i = 0; i < N; ++i)
        printf(" %d ", A[i]);
    printf("\n");
}

int main(int argc, char* argv[])
```

```

{
    int A[N] = {0, 10, 20, 30, 40, 50, 60, 70, 80, 90};
    int tmp;
    print_array(A);

    for(int i = 0; i < N/2; ++i) // Reverse
    {
        tmp = A[i];
        A[i] = A[N-1-i];
        A[N-1-i] = tmp;
    }
    print_array(A);

    tmp = A[0]; // Cycle shift to the left
    for(int i = 0; i < N-1; ++i)
        A[i] = A[i+1];
    A[N-1] = tmp;
    print_array(A);

    tmp = A[N-1]; // Cycle shift to the right
    for(int i = N-1; i > 0; --i)
        A[i] = A[i-1];
    A[0] = tmp;
    print_array(A);

    return 0;
}

```

## Задача №25 ЕГЭ по информатике

Задачи №25 ЕГЭ по информатике на Си



Задача №25 демо-варианта ЕГЭ по информатике 2018 года. Решение на языке Си.

ege25.c

```

#include <stdio.h>
#define N 30

int main() {
    int a[N];
    int i, j, k;

    for (i = 0; i < N; ++i)
        scanf("%d", &a[i]);

    k = 0;
    for (i = 0; i < N; ++i)

```

```
    if (a[i] > 100 && a[i]%5 == 0)
        k++;
    for (i = 0; i < N; ++i)
    {
        if (a[i] > 100 && a[i]%5 == 0)
            a[i] = k;
        printf("%d ", a[i]);
    }

    return 0;
}
```

## Задача №27 ЕГЭ по информатике

### Задача №27 ЕГЭ по информатике на Си



Задача №27 демо-варианта ЕГЭ по информатике 2018 года. Решение на языке Си.

ege27.c

```
#include <stdio.h>

int main()
{
    int N, k26 = 0, k13 = 0, k2 = 0, k1 = 0;
    scanf("%d", &N);

    for(int i = 0; i < N; i++)
    {
        int x;
        scanf("%d", &x);
        if (x%26 == 0)
            k26++;
        else if (x%13 == 0)
            k13++;
        else if (x%2 == 0)
            k2++;
        else
            k1++;
    }

    int m = k26*(k26-1)/2 + k26*(k1+k2+k13) + k2*k13;
    printf("%d\n", m);

    return 0;
}
```

## Добавление и удаление элемента в конец массива

## Добавление и удаление элемента в конец массива на Си



Добавление элемента в конец массива. Удаление элемента в конце массива. Разложение на множители с сохранением их в массиве.

### factorization\_array.c

```
#include <stdio.h>

// returns number of multipliers
// buffer A should be enough for x factors
int get_number_factors(int x, int A[])
{
    int top = 0;
    int divisor = 2;
    while (x != 1)
    {
        while (x%divisor == 0)
        {
            A[top] = divisor;
            top += 1;
            x /= divisor;
        }
        divisor += 1;
    }
    return top;
}

int main(int argc, char* argv[])
{
    int x;
    printf("Enter number to factorize:");
    scanf("%d", &x);
    int A[100];
    int N;
    N = get_number_factors(x, A);

    for(int i = 0; i < N; ++i)
        printf("%d ", A[i]);
    printf("\n");

    return 0;
}
```

## Сортировка массива вставками

## Сортировка массива вставками на Си



Сортировка массива: постановка задачи. Сортировка вставками.

### insert\_sort.c

```
#include <stdio.h>
#include <stdbool.h>
#include <iso646.h>

#define ALLOCATE_SIZE 1000

int input_array(int A[], int max_size)
{
    int top = 0;

    while (true)
    {
        int x;
        scanf("%d", &x);
        if (x == 0 or top == max_size) break;
        A[top] = x;
        top++;
    }
    return top;
}

void print_array(int A[], int N)
{
    for(int i = 0; i < N; ++i)
        printf("%3d ", A[i]);
    printf("\n");
}

void insert_sort(int A[], int N)
{
    for(int i = 1; i < N; ++i)
    {
        int k = i;
        while (k > 0 and A[k-1] > A[k])
        {
            int tmp = A[k-1];
            A[k-1] = A[k];
            A[k] = tmp;
            k -= 1;
        }
    }
}
```



```

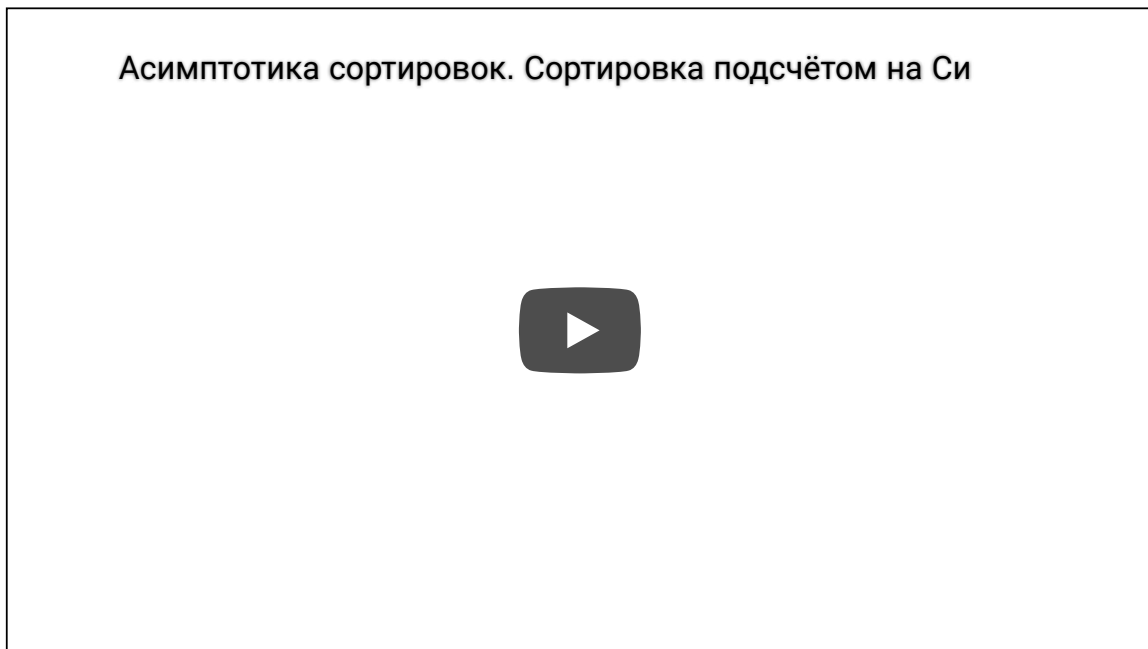
int main(int argc, char* argv[])
{
    printf("Enter numbers:");
    int A[ALLOCATE_SIZE];
    int N;

    N = input_array(A, ALLOCATE_SIZE);
    insert_sort(A, N);
    print_array(A, N);

    return 0;
}

```

## Асимптотика сортировок. Сортировка подсчётом



В чём измеряют скорость работы программы. Наихудший и наилучший случаи. Средний случай. Оценка асимптотики сортировки вставками. Сортировка подсчётом. Частотный анализ. Реализация сортировки подсчётом.

### count\_sort.c

```

#include <stdio.h>
#include <stdbool.h>
#include <iso646.h>

int main(int argc, char* argv[])
{
    int counters[10] = {0};
    int x;

    while (true)
    {
        scanf("%d", &x);
        if (x == 10) break; // Terminator is 10.
        if (x < 0 or x > 9) continue;
        counters[x] += 1;
    }

    for (x = 0; x < 10; ++x)
        for (int i = 0; i < counters[x]; ++i)
            printf("%3d ", x);

    return 0;
}

```

## Самостоятельная работа

Уважаемые студенты!

К 3-му уроку есть домашняя работа в форме конкурса: [ссылка на ДЗ №3](#). Ссылка на неё также находится на главной странице сайта.

Если у вас нет логина и пароля, [зарегистрируйтесь на 1-й конкурс](#), и доступ к остальным вы получите автоматически.

Сайт построен с использованием [Pelican](#). За основу оформления взята тема от [Smashing Magazine](#). Исходные тексты программ, приведённые на этом сайте, распространяются под лицензией [GPLv3](#), все остальные материалы сайта распространяются под лицензией [CC-BY](#).