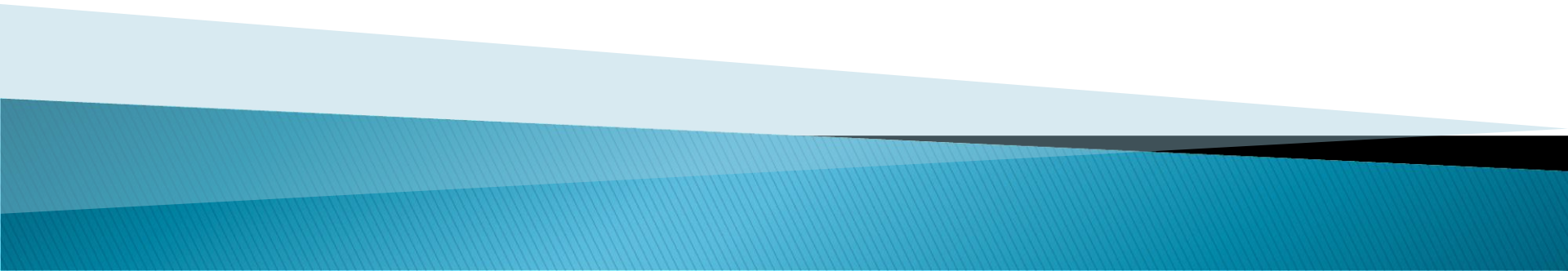


**Саркисян Г.Ф.**

# **ПРОГРАММИРОВАНИЕ**

**Часть I: Структурное программирование**

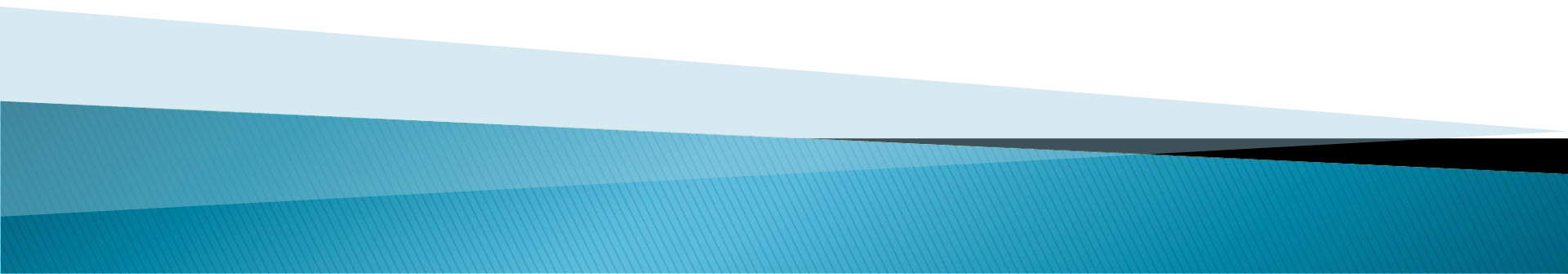


# ОДНОМЕРНЫЙ МАССИВ и УКАЗАТЕЛИ

## Вопросы:

- 1. Определение массива*
- 2. Инициализация массива*
- 3. Ввод вывод просмотр элементов массива*
- 4. Одномерный массив и указатель*
- 5. Передача одномерного массива в функцию*

# ОПРЕДЕЛЕНИЕ МАССИВА



## ОПРЕДЕЛЕНИЕ МАССИВА

**Массив** – это группа расположенных друг за другом в памяти переменных одного типа и имеющих одно общее имя.

**Элементы массива** – это последовательно расположенные ячейки памяти. Каждому элементу массива отводится одна ячейка памяти.

Все элементы имеют одно имя – *имя массива* и отличаются индексами – *порядковыми номерами* в массиве.

## ОПРЕДЕЛЕНИЕ МАССИВА

*Количество элементов* в массиве называется его размером.

Чтобы отвести в памяти нужное количество ячеек для размещения массива, надо заранее знать его размер.

Резервирование памяти для массива выполняется на этапе компиляции программы.

## ОПРЕДЕЛЕНИЕ МАССИВА

Определение массива: **Type** ID\_array [**Size**];

**Type** – это любой тип определенный в программе, кроме *функции*, типа *void* и *ссылки*.

ID\_array (имя массива) – это **скрытый константный указатель на первый элемент массива**.

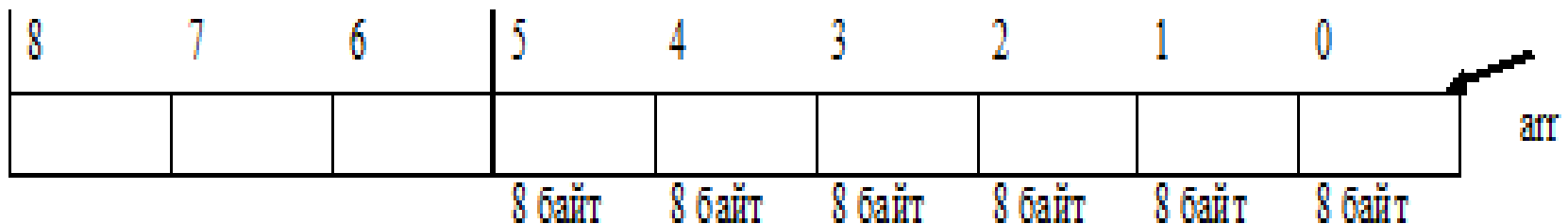
**Size** (размер) для любого типа массива – это **целая положительная константа**, которая может быть задана с помощью константы или константного выражения.

## ОПРЕДЕЛЕНИЕ МАССИВА

Пример определения массива

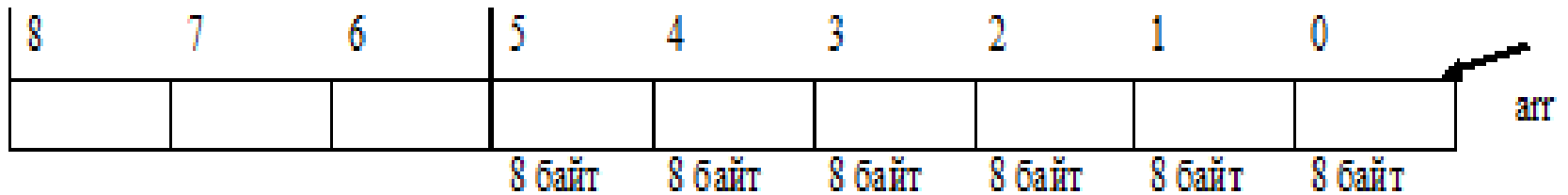
```
const int N = 6; // размер массива
```

```
double arr[N]; //массив вещественных чисел
```



**arr** (имя массива)– это адрес первого элемента массива, т.е. **arr** – это **&arr [0]**

## ОПРЕДЕЛЕНИЕ МАССИВА



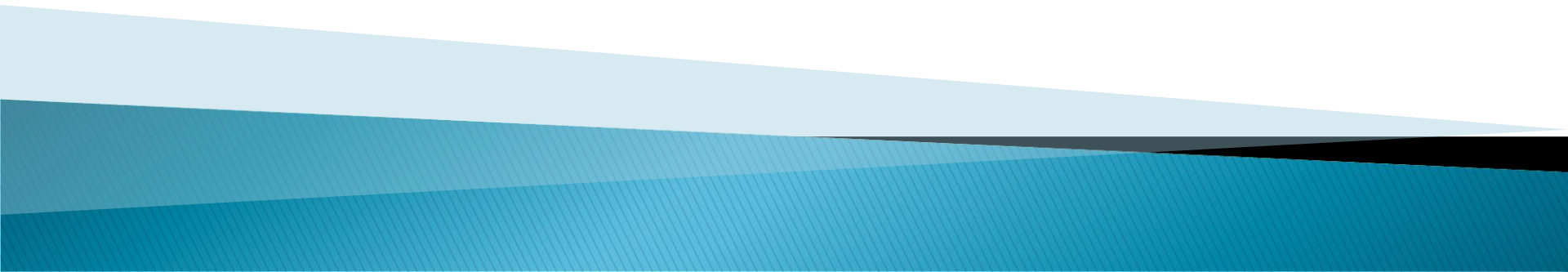
Чтобы обратиться к элементу массива, надо указать имя массива и номер элемента в массиве (индекс):

`arr[ind]` ; // это число вещественного типа

где индекс это целое положительное число, которое может меняться в интервале  $[0, N-1]$ , где  $N$  – размер.



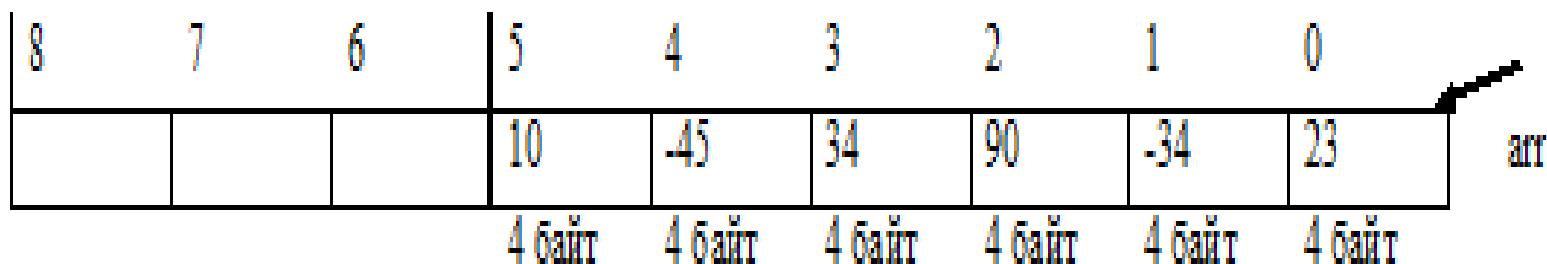
# ИНИЦИАЛИЗАЦИЯ МАССИВА



## ИНИЦИАЛИЗАЦИЯ МАССИВА

1. определение и инициализация массива целых чисел

```
const int N = 6; // размера массива  
int arr[N] = {23, -34, 90, 34, -45, 10};
```



## ИНИЦИАЛИЗАЦИЯ МАССИВА

2.

```
const int N = 6;
```

```
int arr[N] = {23, -34, 90, 34, -45, 10, 23};
```

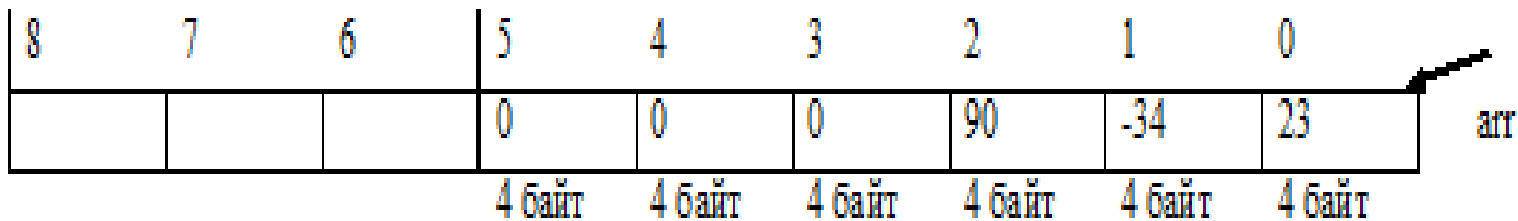
Если количество значений, перечисленных при инициализации больше, чем может поместиться в самом массиве, то при компиляции, программа обнаружит ошибку, т.к. размер статического массива определяется при компиляции.

## ИНИЦИАЛИЗАЦИЯ МАССИВА

3.

```
const int N = 6;
```

```
int arr[N] = {23, -34, 90};
```



Если количество значений, перечисленных при инициализации меньше, то остальное содержание массива автоматически обнуляется.

## ИНИЦИАЛИЗАЦИЯ МАССИВА

4. Обнуление значений массива при его определении:

```
int arr[N] = {0};
```

5. Размер массива может вычисляться компилятором по количеству значений, перечисленных при инициализации.

```
int arr[] = {23, -34, 90, 5, 345};
```

Следовательно, размер данного массива будет равен 5.

## ИНИЦИАЛИЗАЦИЯ МАССИВА

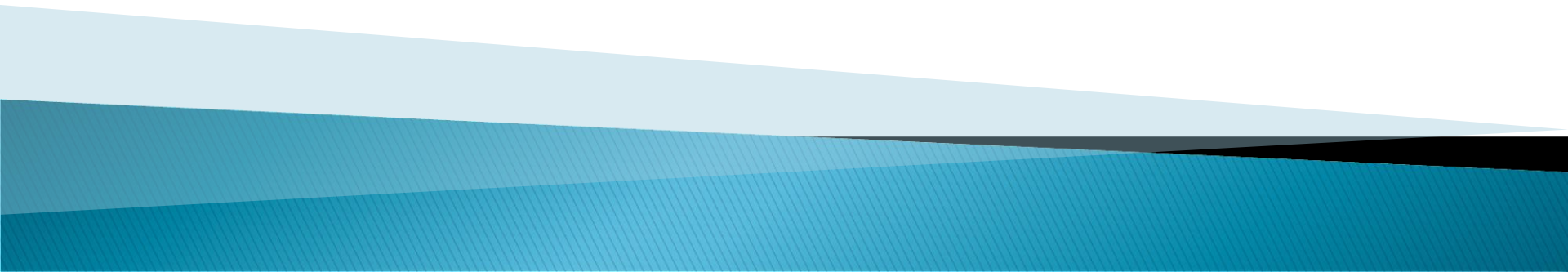
```
int N1, N2;
```

```
double arr[10];
```

```
N1 = sizeof ( arr );
```

```
N2 = sizeof ( arr ) / sizeof ( arr[0] );
```

# **ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА**



## ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

### ВВОД и ВЫВОД ЭЛЕМЕНТОВ МАССИВА

```
const int N = 10;    // размер массива
double x[N];        // массив вещественных чисел
int ind;            //индекс массива

//ввод значений
for (ind = 0; ind < N; ++ind){
    printf("[%d]=" , ind);
    scanf("%lf", &x[ind]);
}

//вывод значений
for(ind = 0; ind < N; ++ind)
    printf("[%d]=%.2lf\n", ind, x[ind]);
```



## ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

### ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

```
const int N = 10;    // размер массива
double x[N];        // массив вещественных чисел
int ind;            //индекс массива
```

#### 1. Слева направо с шагом отличным от 1

```
for(ind = 0; ind < N; ind += step)    {
    обработка x[ind] ;
}
```

## ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

### ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

```
const int N = 10;    // размер массива
double x[N];        // массив вещественных чисел
int ind;            //индекс массива
```

2. Справа налево с шагом 1

```
for(ind = N-1; ind >= 0; ind--) {
    обработка x[ind] ;
}
```

## ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

### ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

```
const int N = 10;    // размер массива
double x[N];        // массив вещественных чисел
int ind;            //индекс массива
```

3. Обработка элементов массива с обеих сторон, пока не дойдем до его середины

```
int j;
for(ind = 0, j = N-1; j > ind; ++ind, --j){
    обработка x[ind] и x[j] ;
}
```

# ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

## ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ МАССИВА

1.

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
void main(void) {
    int a[100], n;
    srand( time( NULL ) );
    printf("\nEnter the size of array: ");
    scanf("%d", &n);
    for(int i = 0; i < n; ++i)    {
        a[i]= 1 + rand()%100; printf("%d ",a[i]);
    }
}
```

## ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

2.

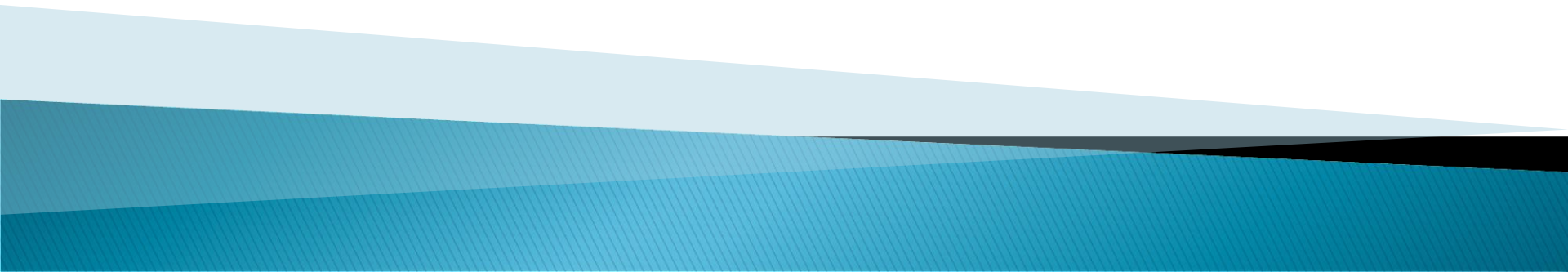
```
void main(void) {  
    const int n = 10;  
    double x[n] = {1.1, 2.2, 3.3, 4.4, 5.5, 6.6,  
                   7.7, 8.8, 9.9, 10.1}, buf;  
  
    int k,i,t;  
    printf("\nK= "); scanf("%d", &k);  
  
    for(t = 0; t < k; ++t) {  
        buf = x[0];  
        for(i = 0; i < n-1; ++i)    x[i] = x[i+1];  
  
        x[n-1] = buf;  
    }  
    for(i = 0; i < n; ++i)    printf("%.11f ", x[i]);  
}
```

## ВВОД ВЫВОД ПРОСМОТР ЭЛЕМЕНТОВ МАССИВА

3.

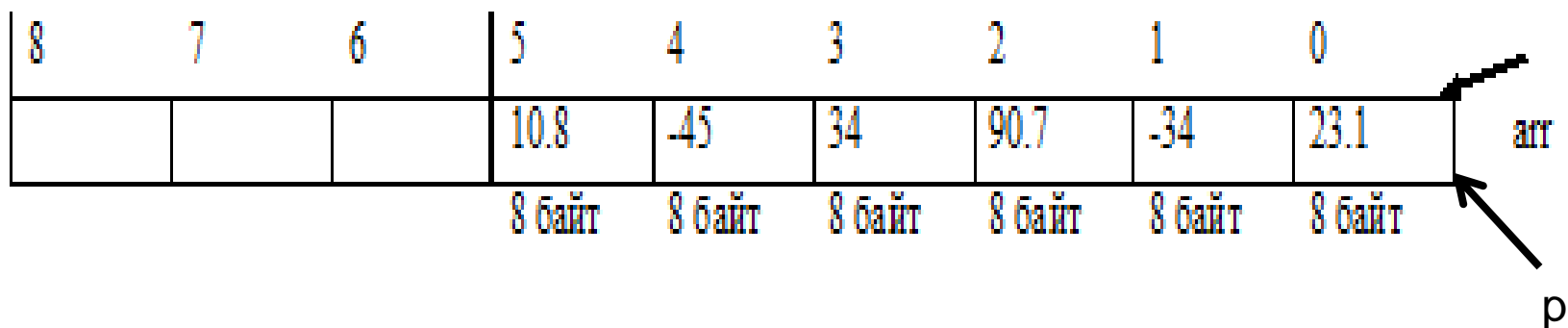
```
void main(void) {  
    const int n = 10;  
    double x[n] = {1.1, -2.2, 3.3, 4.4, 5.5,  
                   68.9, 7.7, 8.8, 9.9, -10.1};  
  
    int i, k ;  
    k = 0;  
    for( i = 1; i < n; ++i)  
        if(x[i] > x[k])    k = i;  
  
    printf("max = [%d] = %.21f\n", k, x[k]);  
}
```

# ОДНОМЕРНЫЙ МАССИВ И УКАЗАТЕЛЬ



## ОДНОМЕРНЫЙ МАССИВ И УКАЗАТЕЛЬ

```
const int N = 6;  
double arr[N] = {23.1, -34, 90.7, 34, -45, 10.8};  
double *p;
```



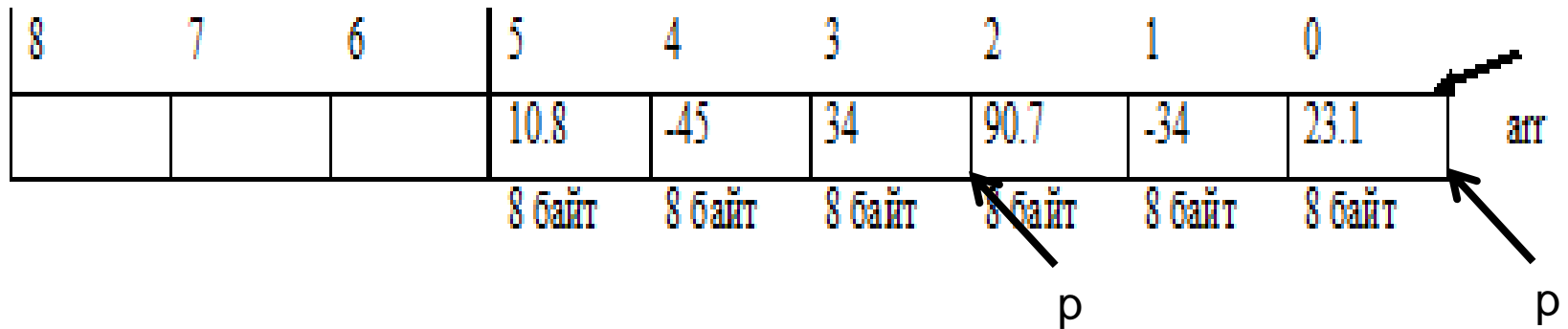
Направить указатель на массив можно двумя способами:

1. `p = arr;`
2. `p = &arr[0];`



## ОДНОМЕРНЫЙ МАССИВ И УКАЗАТЕЛЬ

```
const int N = 6;  
double arr[N] = {23.1, -34, 90.7, 34, -45, 10.8};  
double *p;
```



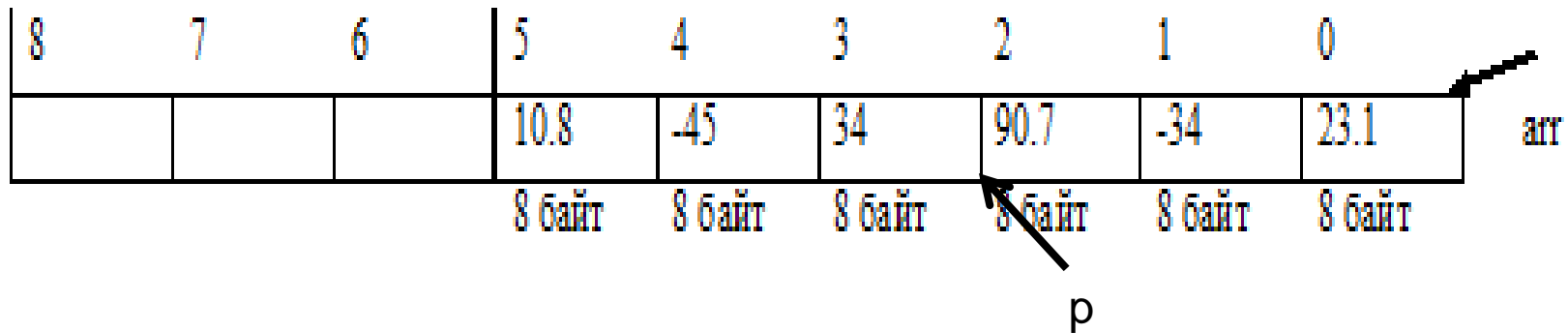
Если `p = p + 3;` // `p + 3*sizeof(double)`

Если на ячейку памяти указывает указатель,  
то:

```
p == &arr[3] == &p[3];  
*p == arr[3] == p[3] == 34;
```

## ОДНОМЕРНЫЙ МАССИВ И УКАЗАТЕЛЬ

```
const int N = 6;  
double arr[N] = {23.1, -34, 90.7, 34, -45, 10.8};  
double *p;
```



```
printf("%.11f\n", *(p+2));
```

После этого действия, куда будет указывать указатель p.

## ОДНОМЕРНЫЙ МАССИВ И УКАЗАТЕЛЬ

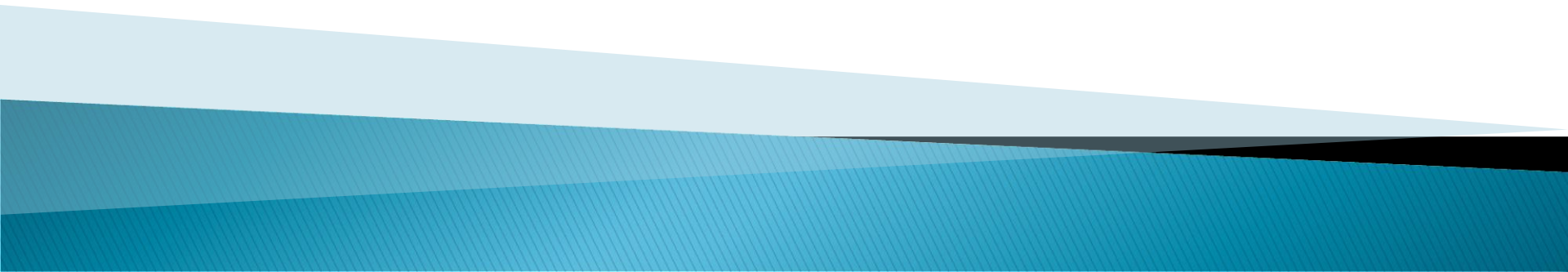
```
int z, a[]={2,3,4}, *p = &a[1];  
z = ++*p--;  
printf("%5d%5d\n", z, *p);
```

## ОДНОМЕРНЫЙ МАССИВ И УКАЗАТЕЛЬ

### ВВОД И ВЫВОД ЭЛЕМЕНТОВ МАССИВА ЧЕРЕЗ УКАЗАТЕЛЬ

```
const int N = 6;
double arr[N];
int ind;
//ввод значений через указатель
    for (ind = 0; ind < N; ++ind) {
        printf("[%d]=", ind);
        scanf("%lf", (arr + ind));
    }
//вывод значений через указатель
for(ind = 0; ind < N; ++ind)
printf("[%d]=%.2lf\n", ind, *(arr + ind));
```

# ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ



## ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ

Передача одномерного массива в функцию имеет вид:

```
ReturnType FunctionID (ArrayType *, int);
```

При вызове функции параметры необходимо передавать следующие образом:

1. первый параметр – это имя массива;
2. второй параметр – это размер массива.

```
#include <stdio.h>
void vvod_mas(double *,int);
void vivod_mas(double *, int);

void main(void){
    const int n = 3,m = 2;
    double  x[n], y[m];
    vvod_mas(x,n);
    vvod_mas(y,m);
    printf("Содержание первого массива:\n");
        vivod_mas(x,n);
    printf("Содержание второго массива:\n");
        vivod_mas(y,m);
}

void vvod_mas(double *p, int k){
    for(int i = 0; i < k; ++i){
        printf("[%d]=",i);
        scanf("%lf", p++);  }
}

void vivod_mas(double *p, int k){
    for(int i = 0;i < k; ++i)
        printf("[%d]=%.2lf\n",i,*(p++));
}
```

## ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ

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

```
void vvod_mas(double *,int) ;
```

```
void vivod_mas(double *, int) ;
```

```
void main(void) {
```

```
    const int n = 5,m = 6;
```

```
    double  x[n], y[m] ;
```

```
    vvod_mas(x,n) ;
```

```
    vvod_mas(y,m) ;
```

```
    printf("Содержание первого массива:\n") ;
```

```
        vivod_mas(x,n) ;
```

```
    printf("Содержание второго массива:\n") ;
```

```
        vivod_mas(y,m) ;
```

```
}
```



## ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ

```
void vvod_mas(double *p, int k){
```

```
    for(int i = 0; i < k; ++i){
```

```
        printf("[%d]=", i);
```

```
        scanf("%lf", p++);
```

```
    }
```

```
}
```

```
void vivod_mas(double *p, int k){
```

```
    for(int i = 0; i < k; ++i)
```

```
        printf("[%d]=%.2lf\n", i, *(p++));
```

```
}
```

## ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ

5.

```
#include <stdio.h>
#define SIZE 100
void vvod_mas(double *, int);
void vivod_mas(double *, int);
void del_elm(double *, int&);

void main( void ){
    double arr [SIZE];      int n;
    printf("Enter size of array: ");
    scanf("%d", &n);
    vvod_mas(arr, n);
    del_elm(arr, n);
    vivod_mas(arr, n);
}
```

## ПЕРЕДАЧА ОДНОМЕРНОГО МАССИВА В ФУНКЦИЮ

```
void del_elm(double *a, int &n1){
    int k = 0, i, j;    double y;
    printf("Vvedite chislo dlya udaleniya: ");
    scanf("%lf", &y);

    for( i = 0; i < n1 - k; ++i){
        if(a[i] == y){
            ++k;
            for(j = i; j < n1 - k; ++j)
                a[j] = a[j+1];

            --i;
        }
    }

    n1 -= k;
}
```

# СПАСИБО ЗА ВНИМАНИЕ!

## ВОПРОСЫ ?

### ОДНОМЕРНЫЙ МАССИВ и УКАЗАТЕЛИ

Автор: Саркисян Гаяне Феликсовна