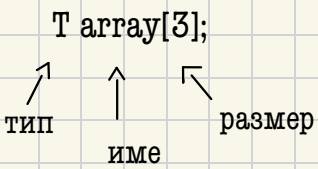
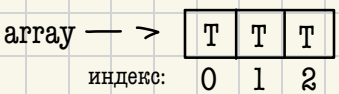
# Тема 6: Масиви с фиксирана дължина

def: Масив - наредена последователност от елементи от един тип.

## 1. Инициализация




Може да се инициализира по следните начини:

```
int array1[3];
int array2 [] = { 1,56,43};
int array3[7] = { 1,56,43};//
останалите елементи се попълват с дефолтна стойност
```

Грешка ще се предизвика, ако се опитаме да инициализираме така:

```
int array4[];
int array5[2] = { 1,56,43};//
декларирали сме, че искаме масив с 2 елемента, а слагаме 3
```

Големината на масива трябва да е константа, чиято стойност да в известна преди компилация.

Пример:

```
int size;
std::cin >> size;
int array[size];//
ГРЕШНО
```

```
const int SIZE = 4;
int array[SIZE]; //✅
```

## 2. Достъп до елемент

Как достъпваме елемент на масива?

- чрез името на масива последвано от индекс на елемент в квадратни скоби

Защо индексацията започва от 0? -
Тъй като array е указател към първия елемент на масива, когато напишем array[2] искаме да каже на указателя да направи 2 отстъпа от началото.

Ако направим следното:

```
std::cout << array;
```

Това ще изпечатва адреса на първия елемент на масива, а не самия масив.

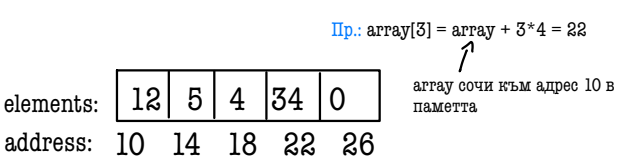
Достъпът до елементите е константен. Това означава, че за приблизително едно и също време ще достъпим елемент на индекс 1 и този на индекс 320 например.

Ако опитаме да достъпим елемент, който не е от масива(памет, която не е наша) => undefined behaviour

Пример:

array[i] = array + i*sizeof(type of array)

int array[5] = {12, 5,4,34,0};
Как примерно би изглеждал масивът в паметта?



## 3. Подаване на масиви във функции

Функцията ще променя ли масива?

- Ако ще променя масива във функцията - int array[].
- Ако няма да го променя - const int array[].

Подаваме масив и размера му.

Прави се копие на указателя.

Пример:

```
void print(const int arr[], int len) {
    for (int i = 0; i < len; i++) {
        std::cout << arr[i] << " ";
    }
}// ако вместо const int arr[] бяхме написали
const int* arr щеше да е еквивалентно

int main() {
    const int SIZE = 4;
    int arr[SIZE]{ 1, 2, 3, 4 };

    print(arr, SIZE);
    return 0;
}
```

Масивите се подават във функциите по адрес =>
Промените, които се правят върху масива във функцията, ще се отразят на подадения масив.

Какво е указател накратко?(засега)

def: Указател - променлива, която пази адрес като стойност.

- Променливата на указател сочи към същия тип данни, от който е и тя, и се създава чрез оператор *.
- Адресът на променливата, с която работим, се присвоява на указателя.

Пример:

```
int number = 5;
int *ptr = &number; //
указател, който пази адреса
на променливата number


std::cout << ptr; //отпечатва
адреса на променливата
number
std::cout << *ptr;//отпечатва
стойността на number
```