Lec-5

Array

Slide-1

Overview (1/2)

- Definition of Array
- Use of Array
- Array Elements Storage in Memory Declaration of Array
- Initialization of Array
- Partial Initialization of Array

Overview (2/2)

- Adjacent Address of Array Elements
- Getting Size of an Array in C
  - Size of Integer Array
  - Size of Double Type Array
  - Size of Character Array
- Accessing Array Elements
- Finding Average Using Array
- Program to Input and Display Elements Using an Array

What is Array?

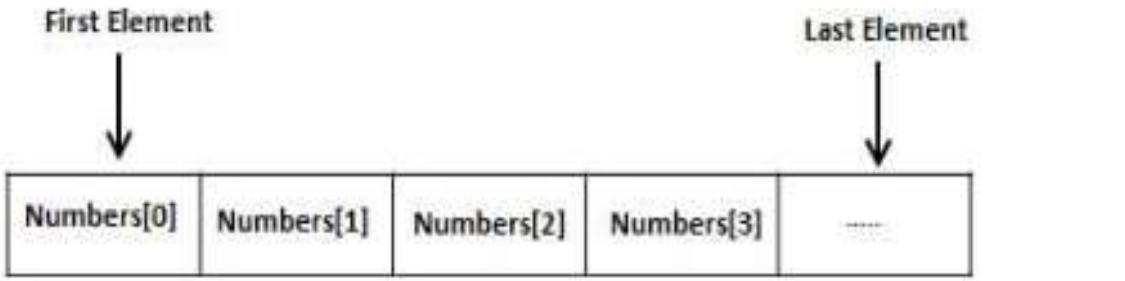
- An array in C is a collection of data items of similar data type.
  - One or more values same data type, which may be primary data types (int, float, char), or user-defined types such as struct or pointers can be stored in an array.
  - In C, the type of elements in the array should match with the data type of the array itself.
- The size of the array, also called the length of the array, must be specified in the declaration itself.
- Once declared, the size of a C array cannot be changed.
- When an array is declared, the compiler allocates a continuous block of memory required to store the declared number of elements

Use of Array

- Arrays are used to store and manipulate the similar type of data.
- Arrays offer a compact and memory-efficient solution.
  - Since the elements in an array are stored in adjacent locations, any element in relation to the current element can be accessed.
  - As each element has an index, it can be directly manipulated.

Storage System

- Array elements are stored in contiguous memory locations.
- Each element is identified by an index starting with "0".
  - The lowest address corresponds to the first element and the highest address to the last element.



Declaration of Array

Code

```c
#include <stdio.h>

int main(){
    int arr[5];
    int i;

    for (i = 0; i <= 4; i++){
       printf("a[%d]: %d\n", i, arr[i]);
    }
    return 0;
}
```

Output

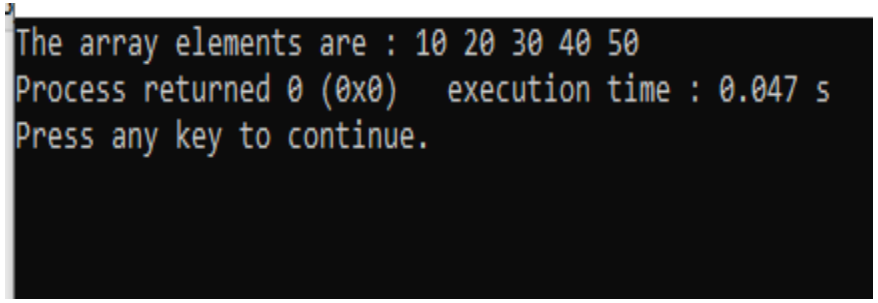
```
a[0]: 8
a[1]: 0
a[2]: 44
a[3]: 0
a[4]: 7279552
```

Initialization of Array

Code

```c
#include <stdio.h>

int main()
{
    int numbers[5] = {10, 20, 30, 40, 50};
    int i;

    printf("The array elements are : ");
      for (i = 0; i < 5; i++) {
        printf("%d ", numbers[i]);
      }

  return 0;
}
```

Output



Initialization of Array

(Initialization of All Array Elements to 0)

Code

```c
#include <stdio.h>

int main(){
    int arr[5] = {0};
    int i;

    for(i = 0; i <= 4; i++){
        printf("a[%d]: %d\n", i, arr[i]);
    }
    return 0;
}
```

Output

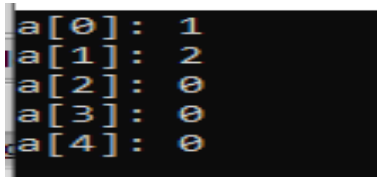
```
a[0]:  0
a[1]:  0
a[2]:  0
a[3]:  0
a[4]:  0
```

Partial Initialization of Array

Code

```c
#include <stdio.h>

int main(){
    int arr[5] = {1,2};
    int i;

    for(i = 0; i <= 4; i++){
        printf("a[%d]: %d\n", i, arr[i]);
    }
    return 0;
}
```

Output

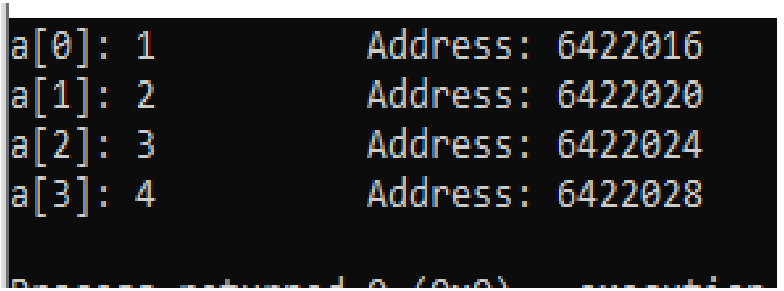


Adjacent Address of Array Elements

Code

```c
#include <stdio.h>

int main(){
   int a[] = {1, 2, 3, 4, 5};
   int i;

   for(i = 0; i < 4; i++){
      printf("a[%d]: %d \t Address: %d\n", i, a[i], &a[i]);
   }
   return 0;
}
```

Output



Getting Size of Array (Integer Array)

Code

```c
#include <stdio.h>

int main(){
    int arr[5] = {1, 2, 3, 4, 5};
    printf("Size of array: %d", sizeof(arr));
    return 0;
}
```

Output

```
Size of array: 20
Process returned 0 (0x0)
Press any key to continue.
```

Getting Size of Array (Double Type Array)

Code

```c
#include <stdio.h>

int main(){
    double a[] = {1.1, 2.2, 3.3, 4.4, 5.5};
    int i;

    for(i = 0; i < 4; i++){
       printf("a[%d]: %f \t Address: %ld\n", i, a[i], &a[i]);
    }
    printf("Size of Array is:%d\n",sizeof(a));
    return 0;
}
```

Output

```
a[0]: 1.100000   Address: 6422000
a[1]: 2.200000   Address: 6422008
a[2]: 3.300000   Address: 6422016
a[3]: 4.400000   Address: 6422024
Size of Array is:40
```

Getting Size of Array (Character Array)

Code

```c
#include <stdio.h>

int  main(){
   char a[] = "Hello";
   int i;

   for (i=0; i<5; i++){
      printf("a[%d]: %c address: %ld\n", i, a[i], &a[i]);
   }
   printf("Size of the Array is:%d\n",sizeof(a));
   return 0;
}
```

Output

```
a[0]: H address: 6422038
a[1]: e address: 6422039
a[2]: l address: 6422040
a[3]: l address: 6422041
a[4]: o address: 6422042
Size of the Array is:6
```

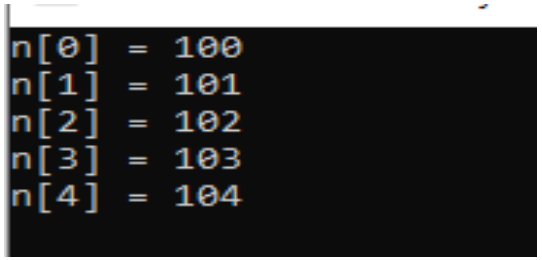
- The size of the character array a[] initialized with the string "hello" is 6 bytes.
- The array a contains the characters 'h', 'e', 'l', 'l', 'o', and a null terminator '\0'.
- The null terminator is automatically added at the end of the string to signify the end of the string in C.

Array Elements Access in C

Code

```c
#include <stdio.h>

int main(){

    int n[5],i,j;

    for(i = 0; i < 5; i++){
        n[i] = i + 100;
    }
    for(j = 0; j < 5; j++){
        printf("n[%d] = %d\n", j, n[j]);
    }
    return 0;
}
```

Output

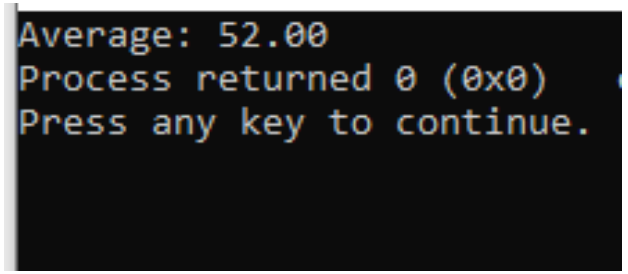


Example
Finding out Average Using Array

Code

```c
#include <stdio.h>

int main(){
    int marks[10] = {50, 55, 67, 73, 45, 21, 39, 70, 49, 51};
    int i, sum = 0;
    float avg;

    for (i = 0; i <= 9; i++){
        sum += marks[i];
    }
    avg = (float)sum / 10;
    printf("Average: %.02f", avg);
    return 0;
}
```

Output

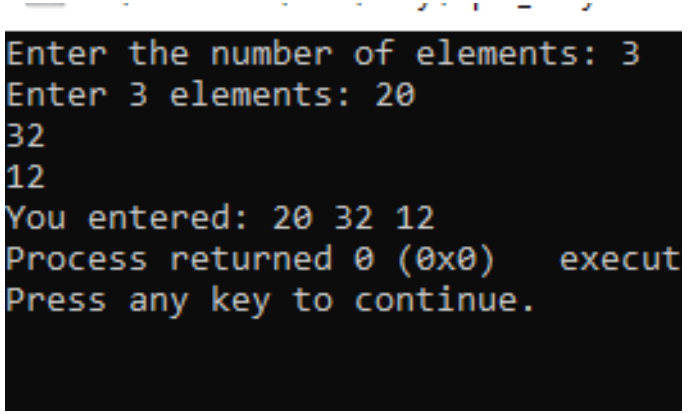


Program to Input and Display Elements Using an Array

Code

```c
#include <stdio.h>

int main() {
    int n;

    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];

    printf("Enter %d elements: ", n);
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("You entered: ");
    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Output



Reference

- Geeksforgeeks
- GrayCode
- Programiz.com
- TutorialsPoint

Thanks