Array

PRESENTED BY: SATYA PRAKASH PATEL
EMAIL: SATYAPATEL.IND@GMAIL.COM

Sometimes we have to deal with Large Amounts of Data

Examples:

1. Marks of all CS 101 students
2. Pressure measured at various points in an area [Weather forecasting]
3. Account balance of thousands of bank customers
4. list of Employee

Array

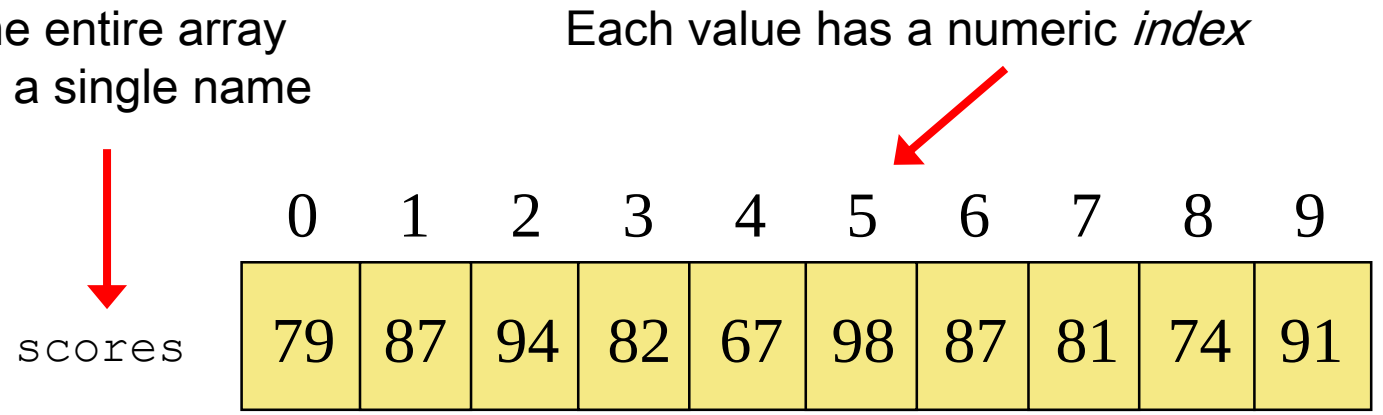
- An array is a collection of data elements that are of the same type (e.g., a collection of integers, collection of characters, collection of doubles).
- arrays are referred to as structured data types. An array is defined as finite, ordered collection of homogenous data, stored in contiguous memory locations.

Arrays

An *array* is an ordered list of values

The entire array has a single name

Each value has a numeric *index*



An array of size N is indexed from zero to N-1

This array holds 10 values that are indexed from 0 to 9

Why do we need arrays?

We can use normal variables (v1, v2, v3, ..) when we have small number of objects, but if we want to store large number of instances, it becomes difficult to manage them with normal variables. The idea of array is to represent many instances in one variable.

Arrays:

A particular value in an array is referenced using the array name followed by the index in brackets

For example, the expression

```
scores[2]
```

refers to the value `94` (the 3rd value in the array)

That expression represents a place to store a single integer and can be used wherever an integer variable can be used

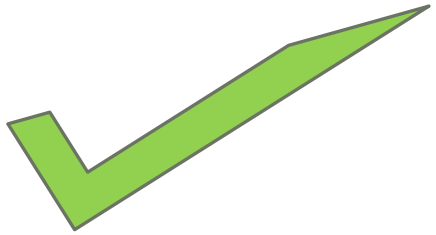
Declaring an Array

```
data-type variable-name[Constant_size];
```

Example

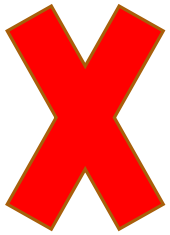
```
int arr[10];
```

Right



```
int m;
int array[m];
```

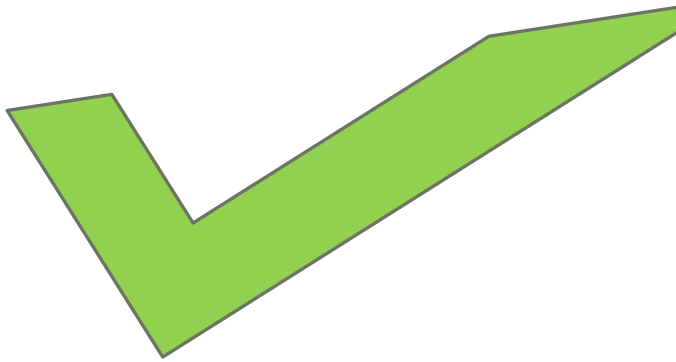
Wrong



Declaring an Array

```
#include<stdio.h>
main()
{int n;
   scanf("%d", &n);
   printf("n=%d\n",n);
   int a[n];
   printf("%d",sizeof(a));
}
```

```
#include<stdio.h>
#define MAX 100
main()
{
int a[MAX];
   printf("%d",sizeof(a));
}
```



Initialization of an Array

Compile time Array initialization

Runtime Array initialization

Memory allocation of array

Stack and a Heap ?

Stack is used for static memory allocation and Heap for dynamic memory allocation, both stored in the computer's RAM .

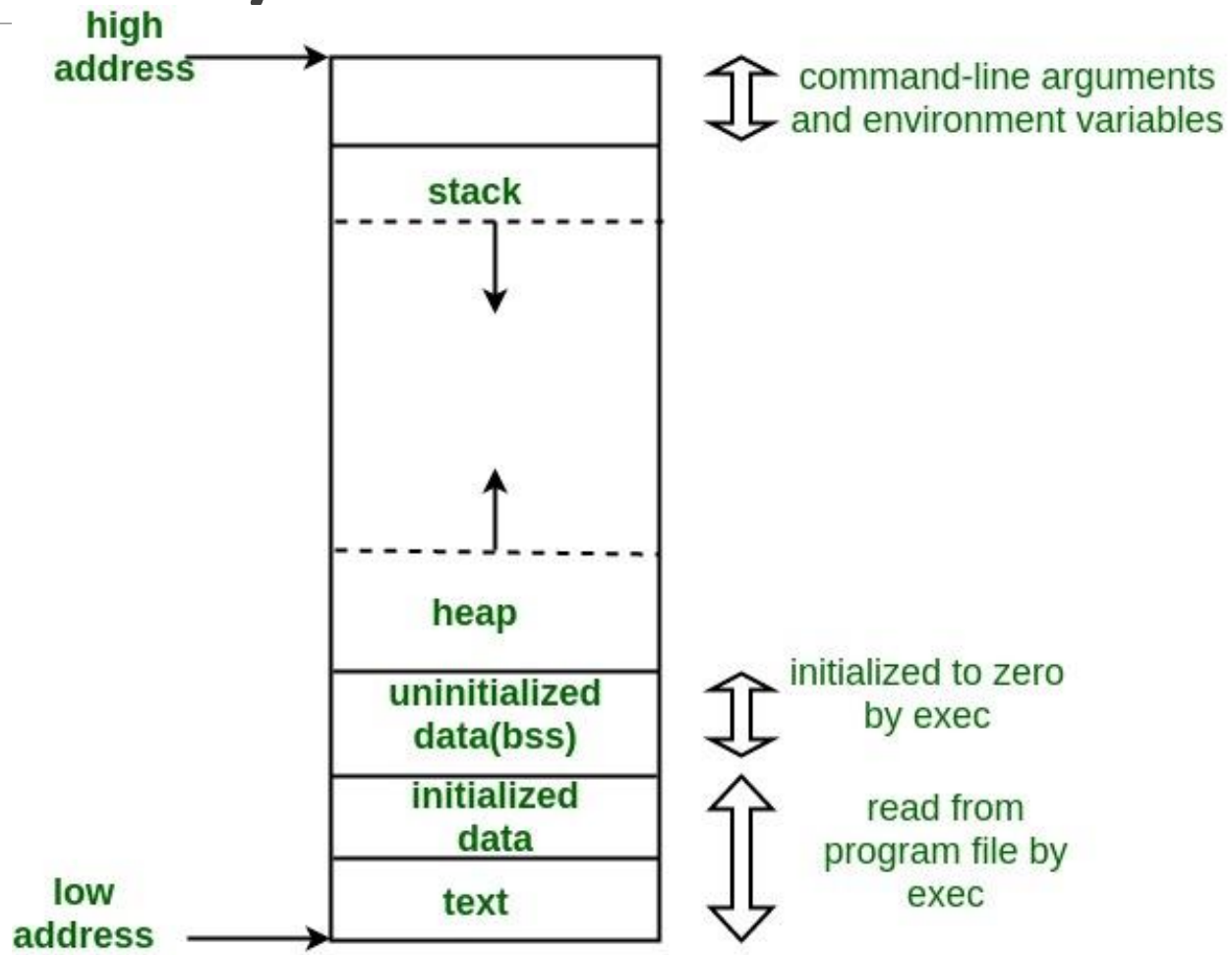


Image Source :
https://www.geeksforgeeks.org/memory-layout-of-c-program/

Multidimensional Array

Run time error on array: Segmentation fault

Question 1

What will be the output of following program ?

Question 2

In which stage the following code #include<stdio.h> gets replaced by the contents of the file stdio.h

A. During editing
B. During linking
C. During execution
D. During pre-processing

Question 3

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include<stdio.h>
int main()
{
        int i, j;
        for(i = 1, j = 1;i<=3,j<=3;i++,j++)
        printf("%d %d ",i, j);
        return 0;
}
```

OPTIONS

A. Compilation Error

B. 1 2 3 1 2 3

C. 1 1 2 2 3 3

D. None of the above

Question 4

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include<stdio.h>
#define loop for(;;)
int main()
{
        printf(“F.R.I.E.N.D.S");
        loop;
        return 0;
}
```

OPTIONS

A. Compilation error

B. F.R.I.E.N.D.S

C. Program never ends

D. None of the above

Question 5

WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM ?

```
#include <stdio.h>

int main()

{int i;
     if (printf("0"))

         Printf(" Coooooooooooool");

    else

         i = 5;

    printf("%d", i);

    return 0;

}
```

OPTIONS

(A) Runtime Error
(B) 0 Coooooooool
(C) Coooooooool

(D) None

Compulsory Homework: Make it Menu driven program

1. Display elements of M X N matrix in matrix format
2. Check if two given matrices are identical
3. To find transpose of a matrix
4. Subtraction of matrices
5. Addition of two matrices
6. Multiplication two matrices
7. Print Matrix Diagonally
8. Given Matrix is an Identity Matrix or not
9. Print upper/lower triangular elements in matrix format
10. Calculate the Sum of the Elements of each Row & Column
11. Find the Frequency of Odd & Even Numbers in the given Matrix
12. Print only border element of a matrix in matrix format rest value must be replaced with space.

Compute Following expressions on matrix, where all the operands are matrix of valid order.

1. A*B-C
2. A*B-C*D
3. A-C*D+B
4. A^3
5. A-B*A^3

Queries and Feedback