Indian Institute of Information Technology, Pune

Department	Computer science and engineering		
Name:	Pranav More	Enrollment No:	112015086
Year:	2nd year	Semester:	4th
Date of Submission	6/3/2022	Subject Name	Java programming

Theory Assignment – 4

Q1	How can declare and use 1 Dim Array? Explain with suitable example.
Ans-1	One Dimensional Array in : One dimensional array is an array that has only one subscript specification that is needed to specify a particular element of an array. A one-dimensional array is a structured collection of components (often called array elements) that can be accessed individually by specifying the position of a component with a single index value. Syntax: data-type arr_name[array_size]; Rules for Declaring One Dimensional Array An array variable must be declared before being used in a program. The declaration must have a data type(int, float, char, double, etc.), variable name, and subscript. The subscript represents the size of the array. If the size is declared as 10, programmers can store 10 elements. An array index always starts from 0. For example, if an array variable is declared as s[10], then it ranges from 0 to 9. Each array element stored in a separate memory location. Initialization of One-Dimensional Array in

An array can be initialized at either following states:

At compiling time (static initialization)

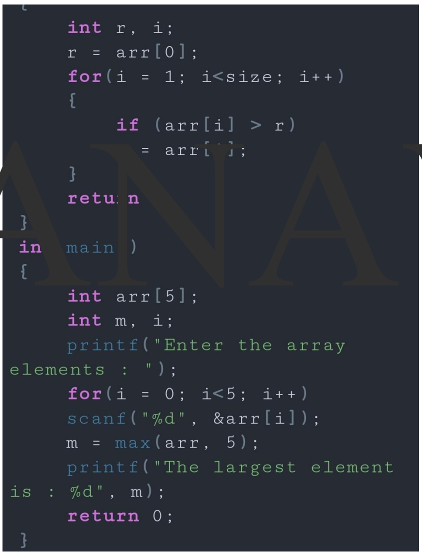
Dynamic Initialization

Compiling time initialization:

The compile-time initialization means the array of the elements are initialized at the time the program is written or array declaration.

Example:

Create an integer array with size 5 and then calculate the larger element of that array using the function



```
 {
     int r, i;
     r = arr[0];
     for(i = 1; i<size; i++)
     {
         if (arr[i] > r)
           r = arr[i];
     }
     return
 }
 int main()
 {
     int arr[5];
     int m, i;
     printf("Enter the array
elements : ");
     for(i = 0; i<5; i++)
     scanf("%d", &arr[i]);
     m = max(arr, 5);
     printf("The largest element
is : %d", m);
     return 0;
 }
```

Output- 82

Q2	How can declare and initialize 2 Dim Array. Explain with suitable example.
Ans-2	Two Dimensional Array in

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure. It provides ease of holding the bulk of data at once which can be passed to any number of functions wherever required.

Declaration of two dimensional Array in
The syntax to declare the 2D array is given below.

```
data_type array_name[rows][columns];
```

Consider the following example.

```
int twodimen[4][3];
```

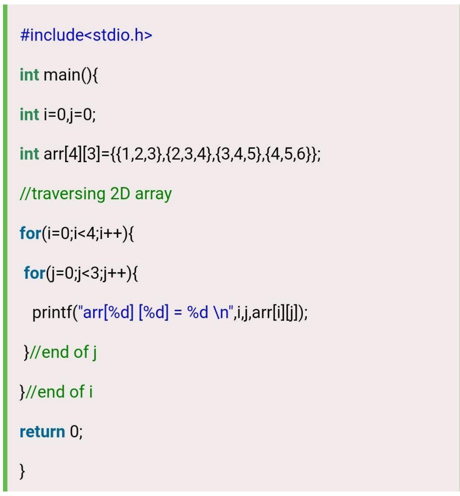
Here, 4 is the number of rows, and 3 is the number of columns.

Initialization of 2D Array in

In the 1D array, we don't need to specify the size of the array if the declaration and initialization are being done simultaneously. However, this will not work with 2D arrays. We will have to define at least the second dimension of the array. The two-dimensional array can be declared and defined in the following way.

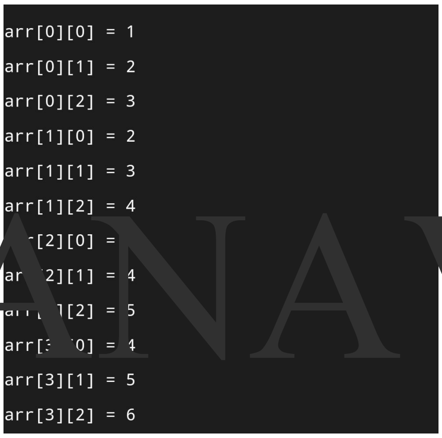
```
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
```

Two-dimensional array example in



```
#include<stdio.h>
int main(){
int i=0,j=0;
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
//traversing 2D array
for(i=0;i<4;i++){
 for(j=0;j<3;j++){
   printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
 }//end of j
}//end of i
return 0;
}
```

Output -

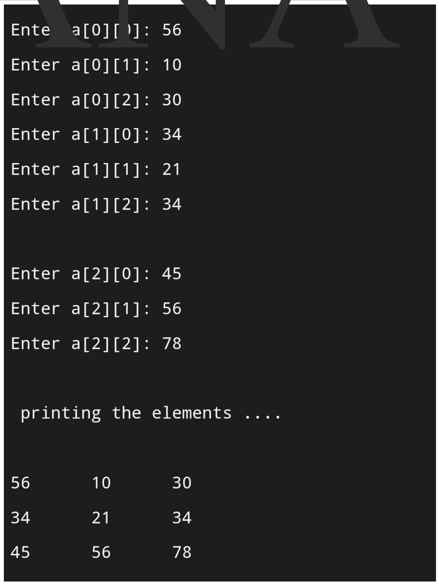


```
arr[0][0] = 1
arr[0][1] = 2
arr[0][2] = 3
arr[1][0] = 2
arr[1][1] = 3
arr[1][2] = 4
arr[2][0] = 3
arr[2][1] = 4
arr[2][2] = 5
arr[3][0] = 4
arr[3][1] = 5
arr[3][2] = 6
```

C 2D array example: Storing elements in a matrix and printing it.

```
#include <stdio.h>
void main ()
{
    int arr[3][3],i,j;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3;j++)
        {
            printf("Enter a[%d][%d]: ",i,j);
            scanf("%d",&arr[i][j]);
        }
    }
    printf("\n printing the elements ....\n");
    for(i=0;i<3;i++)
    {
        printf("\n");
        for (j=0;j<3;j++)
        {
            printf("%d\t",arr[i][j]);
        }
    }
}
```

Output



```
Enter a[0][0]: 56
Enter a[0][1]: 10
Enter a[0][2]: 30
Enter a[1][0]: 34
Enter a[1][1]: 21
Enter a[1][2]: 34

Enter a[2][0]: 45
Enter a[2][1]: 56
Enter a[2][2]: 78

 printing the elements ....


56      10      30
34      21      34
45      56      78
```

Q3	First explain and WAP program to find mean, mode and median. Take a suitable data set.
Ans-3	

Q4	First explain and WAP to find the sum of diagonal elements of a matrix.

Ans-4

This program allows the user to enter the number of rows and columns of a Matrix. Next, we are going to calculate the sum of diagonal elements in this matrix using For Loop.

```
/* Program to find Sum of Diagonal Elements of a Matrix */

#include<stdio.h>

int main()
{
int i, j, rows, columns, a[10][10], Sum = 0;

printf("\n Please Enter Number of rows and columns  :  ");
scanf("%d %d", &i, &j);

printf("\n Please Enter the Matrix Elements \n");
for(rows = 0; rows < i; rows++)
{
for(columns = 0;columns < j;columns++)
{
scanf("%d", &a[rows][columns]);
}
}
for(rows = 0; rows < i; rows++)
{
Sum = Sum + a[rows][rows];
}

printf("\n The Sum of Diagonal Elements of a Matrix
= %d", Sum );

return 0;
}
```

C Program to find Sum of Diagonal Elements of a Matrix 1 In this program to find Sum of Diagonal Elements of a Matrix example

Multiplication of size of 10 * 10. The below statements ask the User to enter the Matrix size (Number of rows and columns. For instance 2 Rows, 3 Columns = a[2][3])

```
printf("\n Please Enter Number of rows and columns :
");
scanf("%d %d", &i, &j);
```

Next, we used Programming for loop to iterate every cell present in a[3][3] matrix. Conditions inside the for loops ((rows < i) and (columns < j)) will ensure the program compiler, not to exceed the Matrix limit. Otherwise, the matrix will overflow. The scanf statement inside the for loop will store the user entered values in every individual array element such as a[0][0], a[0][1], …..

```
for(rows = 0; rows < i; rows++).
```

```
{
for(columns = 0; columns < j; columns++)
{
scanf("%d", &a[rows][columns]);
}
}
```

In the next line, We have one more for loop to find Sum of Diagonal Elements of a Matrix

```
for(rows = 0; rows < i; rows++)
{
        Sum = Sum + a[rows][rows];
}
```

User inserted values for C Program to find Sum of Diagonal Elements of a Multi-Dimensional Array example are: a[3][3] = {{10, 20, 30}, { 40, 50, 60}, {70, 80, 90}}

Row First Iteration: for(rows = 0; rows < 3; 0++)

The condition (0 < 3) is True.

Sum = Sum + a[rows][rows]

Sum = Sum + a[0][0] => 0 + 10 = 10

Row Second Iteration: for(rows = 1; rows < 3; 1++)

The condition (1 < 3) is True.

Sum = Sum + a[1][1]

Sum = 10 + 50 = 60

Row Second Iteration: for(rows = 2; rows < 3; 2++)

The condition (2 < 3) is True.

Sum = Sum + a[2][2]

Sum = 60 + 90 = 150

	Next, rows value will increment. After the increment, the condition inside the for loop (rows < 3) will fail. So it will exit from the loop. At last, we used the printf statement to print the total Sum as output

PRANAV MORE