**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: BTech Integrated Sem III

**Course: C Programming**

**2022-2023**

**Experiment No.07**

PART A

(PART A: TO BE REFFERED BY STUDENTS)

A.1 Aim: **To study Arrays in C Programming.**

P1: Write a program to accept ‘n’ integers from the user into an array and display the count of even and odd numbers of these.

P2: Write a program to find the largest of ‘n’ numbers taken from the user.

P3: Write a program to multiply each element of an array by 5 and display the resultant array.

P4.Write a program to find and display odd & even numbers from an array (1D) separately.

P5.Write a program to copy one 1D array into another 1D array and display copied array.

P6.Implement a program to reverse elements of 1D array and display it.

P7.Write a menu driven program to insert or delete a user specified element from an array and print the resultant array.

P8: Program to calculate average of an array

P9: Develop a program to perform sum of elements of matrix (2D array) of order MXN.

P10:Develop a program to find sum of elements of lower triangular matrix of order MxN.

A.2 Prerequisite:

Knowledge of derived data type like arrays.

A.3 Outcome:

After successful completion of this experiment students will be able to

1. Implement C programs using various derived data type like arrays.

A.4 Theory:

In C programming, one of the frequently arising problem is to handle similar types of data. For example: If the user want to store marks of 100 students. This can be done by creating 100 variable individually but, this process is rather tedious and impracticable. These type of problem can be handled in C programming using arrays.

An array is a sequence of data item of homogeneous value(same type).

Arrays are of two types:

* One-dimensional arrays
* [Multidimensional arrays](http://www.programiz.com/c-programming/c-multi-dimensional-arrays)

General declaration of array is as follows:-

**data\_type array\_name[array\_size];**

**For example**:

int age[5];

Here, the name of array is age. The size of array is 5,i.e., there are 5 items(elements) of array age. All element in an array are of the same type (int, in this case).

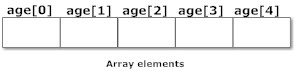
**Array elements**

Size of array defines the number of elements in an array. Each element of array can be accessed and used by user according to the need of program. For example:

int age[5];

Note that, the first element is numbered 0 and so  on.

Here, the size of array age is 5 times the size of int because there are 5 elements. i.e 10 bytes.



Suppose, the starting addres of age[0] is 2120d and the size of int be 2 bytes. Then, the next address (address of a[1]) will be 2122d, address of a[2] will be 2124d and so on.

Arrays can be initialized at declaration time in  this source code as:

int age[5]={2,4,34,3,4};

It is not necessary to define the size of arrays during initialization.

int age[]={2,4,34,3,4};

In this case, the compiler determines the size of array by calculating the number of elements of an array.



/\* C program to find the sum marks of n students using arrays \*/

#include <stdio.h>

int main(){

int marks[10],i,n,sum=0;

printf("Enter number of students: ");

scanf("%d",&n);

for(i=0;i<n;++i){

printf("Enter marks of student%d: ",i+1);

scanf("%d",&marks[i]);

sum+=marks[i];

}

printf("Sum= %d",sum);

return 0;

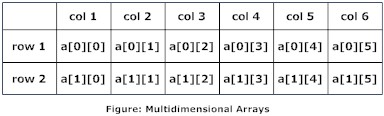
}

C programming language allows to create arrays of arrays known as multidimensional arrays. For example:

float a[2][6];

Here, *a* is an array of two dimension, which is an example of multidimensional array. This array has 2 rows and 6 columns

For better understanding of multidimensional arrays, array elements of above example can be thought of as below:



**Write a C program to find sum of two matrix of order 2\*2 using multidimensional arrays where, elements of matrix are entered by user.**

 #include <stdio.h>

int main()

{

float a[2][2], b[2][2], c[2][2];

int i,j;

printf("Enter the elements of 1st matrix\n “);

for(i=0;i<2;++i)

for(j=0;j<2;++j)

{ printf("Enter a%d%d: ",i+1,j+1);

scanf("%f",&a[i][j]);

}

printf("Enter the elements of 2nd matrix\n");

for(i=0;i<2;++i)

for(j=0;j<2;++j)

{ printf("Enter b%d%d: ",i+1,j+1);

scanf("%f",&b[i][j]);

}

for(i=0;i<2;++i)

for(j=0;j<2;++j)

{

c[i][j]=a[i][j]+b[i][j]; /\* Sum of corresponding elements of two arrays. \*/

}

printf("\nSum Of Matrix:");

for(i=0;i<2;++i)

for(j=0;j<2;++j)

{ printf("%.1f\t",c[i][j]);

if(j==1) /\* To display matrix sum in order. \*/ printf("\n");

}

return 0;

}

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

**(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)**

|  |  |
| --- | --- |
| Roll No. | Name: |
| Program: | Division: |
| Semester: | Batch : |
| Date of Experiment: | Date of Submission: |
| Grade : |  |

B.1 Algorithm

B.2 Flow Chart

B.3 Software Code written by student:

***(Paste your C++ code completed during the 2 hours of practical in the lab here)***

B.4 Input and Output:

***(Paste your program input and output in following format. If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

B.5 Conclusion:

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***