

LAB EXERCISE #5

Objective(s):

To understand programming using different dimensions of Array.

Program: Write a program to insert 5 elements into an array and print the elements of the array.

Code: *(Use comments wherever applicable)*

```
#include<stdio.h>
#include<conio.h>

void main()
{
    int i, arr[5];
    printf("Enter the elements into the array:");
    for(i=0; i<=4;i++)
        scanf("%d",&arr[i]);
    printf("The elements of the array are:");
    for(i=0; i<=4;i++)
        printf("%d \t", arr[i]);
    getch();
}
```

SAMPLE PROGRAMS

(Students are to code the following programs in the lab and show the output to instructor/course Teacher)

Instructions

- Write comment to make your programs readable.
- Use descriptive variables in your programs(Name of the variables should show their purposes)

Programs List

1. Write a C program to find an item from an array by Linear search.
2. Write a Program to perform addition of all elements in Array.
3. Write a Program to find the largest and smallest element in Array.

4. Write a Program to reverse the array elements.
5. Write a C program to insert an element into an array.
6. Write a Program for deletion of an element from the specified location from Array.
7. Write a C program to find an item from an array by binary search
8. Write a program for addition of two matrices.
9. Write a C program to Subtract one matrix from another.
10. Write a C program to transpose a matrix.
11. Write a Program to multiply two 3 X 3 Matrices.
12. Write a program that reads a number **N**. This **N** is the size of a array **X[N]**. Next, read each of the numbers of **X**, find the smallest element of this array and its position within the array, printing this information.
13. Read a number and make a program which puts this number in the first position of an array **N[10]**. In each subsequent position, put the double of the previous position. For example, if the input number is 1, the array numbers must be 1,2,4,8, and so on.

Input

The input contains an integer number **V** ($V < 50$).

Output

Print the stored number of each array position, in the form "**N[i] = X**", where **i** is the position of the array and **x** is the stored number at the position **i**. The first number for **X** is **V**.

Input Sample	Output Sample
1	N[0] = 1 N[1] = 2 N[2] = 4 ...

