**Assigment- 2 Date: 06/02/2024**

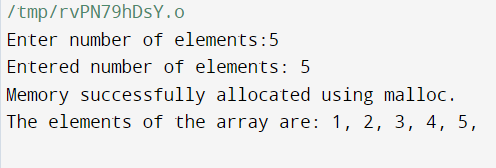
**Pratical-1:**

**Aim: Explain the program and show the output.**

**Code:**

**#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
  
    // This pointer will hold the  
    // base address of the block created  
    int\* ptr;  
    int n, i;  
  
    // Get the number of elements for the array  
    printf("Enter number of elements:");  
    scanf("%d",&n);  
    printf("Entered number of elements: %d\n", n);  
  
    // Dynamically allocate memory using malloc()  
    ptr = (int\*)malloc(n \* sizeof(int));  
  
    // Check if the memory has been successfully  
    // allocated by malloc or not  
    if (ptr == NULL) {  
        printf("Memory not allocated.\n");  
        exit(0);  
    }  
    else {  
  
        // Memory has been successfully allocated  
        printf("Memory successfully allocated using malloc.\n");  
  
        // Get the elements of the array  
        for (i = 0; i < n; ++i) {  
            ptr[i] = i + 1;  
        }  
  
        // Print the elements of the array  
        printf("The elements of the array are: ");  
        for (i = 0; i < n; ++i) {  
            printf("%d, ", ptr[i]);  
        }  
    }  
  
    return 0;  
}**

**Output:**



**Pratical-2:**

**Aim:Write a program to create a dynamic array and enter the values and then display them.**

**Code:**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**// This pointer will hold the**

**// base address of the block created**

**int\* ptr;**

**int n, i;**

**// Get the number of elements for the array**

**printf("Enter number of elements:");**

**scanf("%d",&n);**

**printf("Entered number of elements: %d\n", n);**

**// Dynamically allocate memory using malloc()**

**ptr = (int\*)malloc(n \* sizeof(int));**

**// Check if the memory has been successfully**

**// allocated by malloc or not**

**if (ptr == NULL) {**

**printf("Memory not allocated.\n");**

**exit(0);**

**}**

**else {**

**// Memory has been successfully allocated**

**printf("Memory successfully allocated using malloc.\n");**

**// Get the elements of the array**

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

**ptr[i] = i + 1;**

**}**

**// Print the elements of the array**

**printf("The elements of the array are: ");**

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

**printf("%d, ", ptr[i]);**

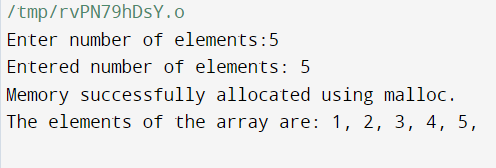
**}**

**}**

**return 0;**

**}**

**Output:**



**Pratical-3:**

**Aim:** **Write a program to add two dynamic array.**

**Code:**

**#include <stdio.h>**

**#include <malloc.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int i = 0;**

**int size = 0;**

**int\* dynamicArray1;**

**int\* dynamicArray2;**

**int\* dynamicArray3;**

**printf("Enter the size for dynamic arrays: ");**

**scanf("%d", &size);**

**dynamicArray1 = (int\*)malloc(size \* sizeof(int));**

**dynamicArray2 = (int\*)malloc(size \* sizeof(int));**

**dynamicArray3 = (int\*)malloc(size \* sizeof(int));**

**printf("Enter Elements of First Array: ");**

**for (i = 0; i < size; i++)**

**scanf("%d", dynamicArray1 + i);**

**printf("Enter Elements of Second Array: ");**

**for (i = 0; i < size; i++)**

**scanf("%d", dynamicArray2 + i);**

**//Now add both arrays**

**for (i = 0; i < size; i++)**

**\*(dynamicArray3 + i) = \*(dynamicArray1 + i) + \*(dynamicArray2 + i);**

**printf("Result of Addition: \n");**

**for (i = 0; i < size; i++)**

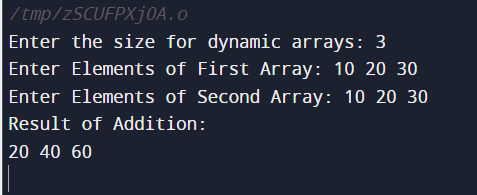
**printf("%d ", \*(dynamicArray3 + i));**

**printf("\n");**

**return 0;**

**}**

**Output:**



**Pratical-4:**

**Aim:** **Write a program to find the greatest value from a dynamic array created during runtime.**

**Code:**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main() {**

**int n;**

**double \*data;**

**printf("Enter the total number of elements: ");**

**scanf("%d", &n);**

**// Allocating memory for n elements**

**data = (double \*)calloc(n, sizeof(double));**

**if (data == NULL) {**

**printf("Error!!! memory not allocated.");**

**exit(0);**

**}**

**// Storing numbers entered by the user.**

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

**printf("Enter number%d: ", i + 1);**

**scanf("%lf", data + i);**

**}**

**// Finding the largest number**

**for (int i = 1; i < n; ++i) {**

**if (\*data < \*(data + i)) {**

**\*data = \*(data + i);**

**}**

**}**

**printf("Largest number = %.2lf", \*data);**

**free(data);**

**return 0;**

**}**

**Output:**

