Experiment no – 06(a)

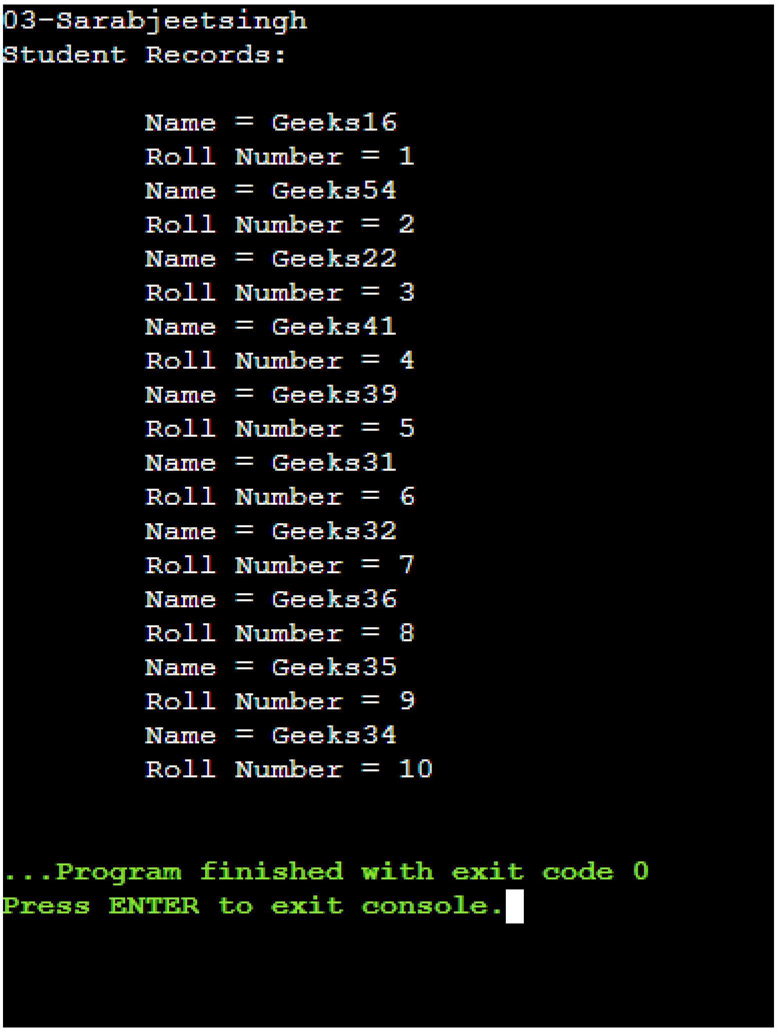
Aim: a. Write a program to print rollno and names of 10 students using array.

Algorithm:

i. Start
ii. Store Student Information
iii. Create the student's structure variable
iv. Display information
v. Stop **Code:**

```
// C Program to Store Information
// of Students Using Structure
#include <stdio.h>
#include <stdlib.h>
#include <string.h> // Create
the student structure struct
Student {
        char* name;
int roll_number;
};
// Driver code int
main()
{  printf("03-sarabjeetsingh\n");
int i = 0, n = 10;
        // Create the student's structure variable
        // with n Student's records
struct Student student[n];      // Get
the students data
student[0].roll_number = 1;
student[0].name = "Geeks16";
student[1].roll_number = 2;
student[1].name = "Geeks54";
student[2].roll_number = 3;
student[2].name = "Geeks22";
student[3].roll_number = 4;
student[3].name = "Geeks41";
student[4].roll_number = 5;
student[4].name = "Geeks39";
student[5].roll_number = 6;
student[5].name = "Geeks3”;
    student[6].roll_number = 7;
            student[6].name = "Geeks32";
student[7].roll_number = 8;  student[7].name =
"Geeks36";
            student[8].roll_number = 9;
            student[8].name = "Geeks35";
            student[9].roll_number = 10;
student[9].name = "Geeks34";          //
Print the Students information
printf("Student Records:\n\n");          for
(i = 0; i < n; i++) {
                        printf("\tName = %s\n", student[i].name);
printf("\tRoll Number = %d\n", student[i].roll_number);
            }
            return 0;
}
```

Output:



```
03-Sarabjeetsingh
Student Records:

        Name = Geeks16
        Roll Number = 1
        Name = Geeks54
        Roll Number = 2
        Name = Geeks22
        Roll Number = 3
        Name = Geeks41
        Roll Number = 4
        Name = Geeks39
        Roll Number = 5
        Name = Geeks31
        Roll Number = 6
        Name = Geeks32
        Roll Number = 7
        Name = Geeks36
        Roll Number = 8
        Name = Geeks35
        Roll Number = 9
        Name = Geeks34
        Roll Number = 10


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment no – 06(b) Aim:

Write a program to read a matrix of size m*n.

Algorithm:

i. Start
ii. Enter row and column size
iii. Construct Matrix
iv. Display result
v. Stop **Code:** *#include<stdio.h> int main()*

{ printf("03-sarabjeetsingh.\n");

int i,j,m,n;

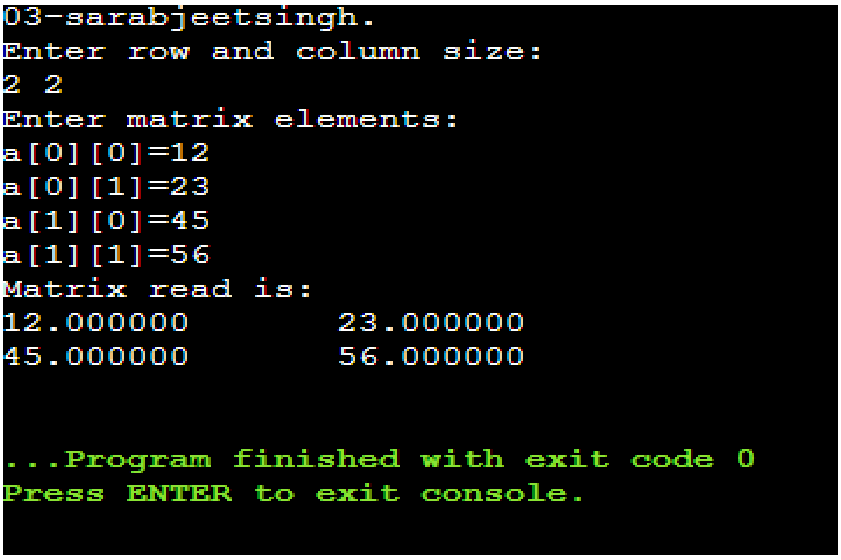
float a[10][10];

printf("Enter row and column size:\n");

scanf("%d%d", &m, &n); printf("Enter

```
matrix elements:\n");  for(i=0;i<
m;i++)
 {
  for(j=0;j<    n;j++)        {
printf("a[%d][%d]=",i,j);
scanf("%f", &a[i][j]);
  }
 }
 printf("Matrix read is:\n");
for(i=0;i< m;i++)
 {
  for(j=0;j< n;j++)
  {
   printf("%f\t",a[i][j]);
  }
printf("\n");
 }
}
```

Output:



```
03-sarabjeetsingh.
Enter row and column size:
2 2
Enter matrix elements:
a[0][0]=12
a[0][1]=23
a[1][0]=45
a[1][1]=56
Matrix read is:
12.000000       23.000000
45.000000       56.000000


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment no – 06(c)

Aim: **Write a program to sort the elements of array in ascending or descending order.**

Algorithm:

i. Start.
ii. Input size of array. iii. Place currently selected element array to its correct place. **iv.** Swap if currently selected array element to its correct place.
v. Print the sorted array.
vi. Stop.

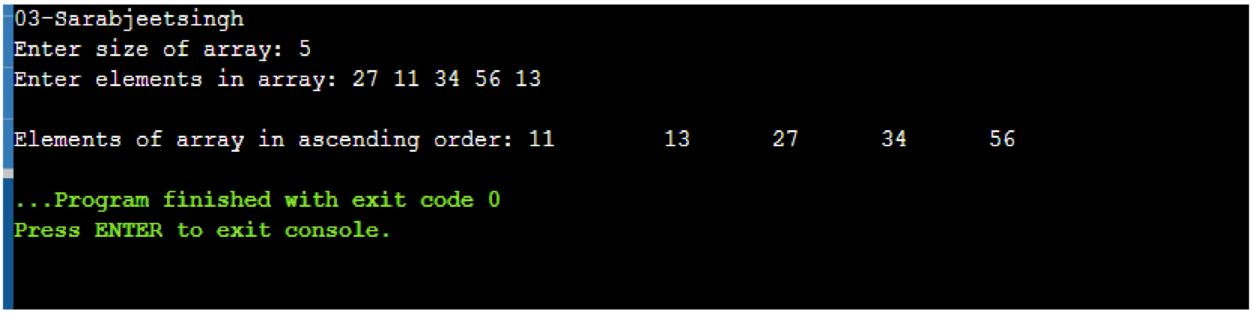
Code:

```
/**
 * C program to sort elements of array in ascending order
 */
#include <stdio.h>
#define MAX_SIZE 100    // Maximum array size
int main()
{ printf(“03-sarabjeetsingh.\n");
int arr[MAX_SIZE];
    int size;
int i, j, temp;
    /* Input size of array */
printf("Enter size of array: ");
scanf("%d", &size);     /* Input
elements in array */     printf("Enter
elements in array: ");     for(i=0;
i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    for(i=0; i<size; i++)
    {
        /*
*          Place currently selected element array[i]
* to its correct place.
       */
      for(j=i+1; j<size; j++)
      {
         /*
*          Swap if currently selected array element
* is not at its correct position.
          */
if(arr[i] > arr[j])
         {               temp
= arr[i];                arr[i]
= arr[j];                arr[j]
= temp;
         }
      }
   }
   /* Print the sorted array */
   printf("\nElements of array in ascending order: ");
for(i=0; i<size; i++)
   {
      printf("%d\t", arr[i]);
   }

   return 0;
}
```

Output:



```
03-Sarabjeetsingh
Enter size of array: 5
Enter elements in array: 27 11 34 56 13

Elements of array in ascending order: 11        13      27      34      56

...Program finished with exit code 0
Press ENTER to exit console.
```