CORE PRACTICALS

1. Write a program to convert temperature from Celsius to Fahrenheit by taking input from the user.

CODE:

```
#include <stdio.h>
#include<conio.h>
int main()
{
   float celsius, fahrenheit;
   printf("Enter temperature in Celsius: ");
   scanf("%f", &celsius);
   fahrenheit = (celsius * 9 / 5) + 32;
   printf("%.2f Celsius = %.2f Fahrenheit", celsius, fahrenheit);
   getch( );
}
```

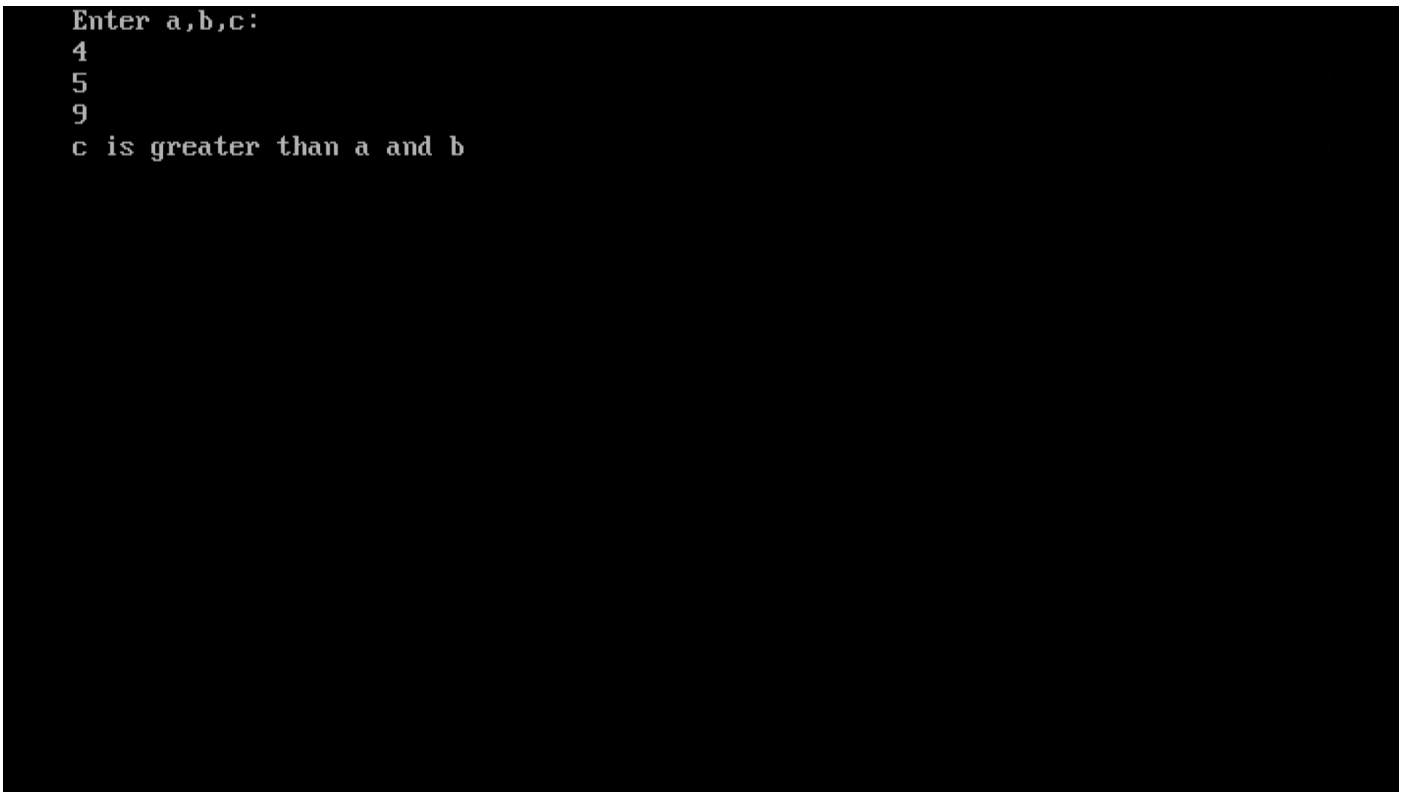
OUTPUT:

2. **Write a program to find the greatest number among three numbers given by the user.**

CODE:

```
#include <stdio.h>
#include<conio.h>
int main()
{

   int a, b, c;
   printf("Enter a,b,c: \n");
   scanf("%d %d %d", &a, &b, &c);

   if (a > b && a > c) {
      printf("a is Greater than b and c");
   }
   else if (b > a && b > c) {
      printf("b is Greater than a and c");
   }
   else if (c > a && c > b) {
      printf("c is Greater than a and b");
   }
   else {
      printf("all are equal or any two values are equal");
   }
   getch( );
   return 0;
}
```

OUTPUT:



3. Write a program to check if a given number is a prime number or not.

CODE:

```
#include <stdio.h>
#include <conio.h>
int main( )
{
  int n, i, c = 0;
  printf("Enter any number n:");
  scanf("%d", &n);
  for (i = 1; i <= n; i++)
{
     if (n % i == 0)
    {
       c++;
     }
  }

  if (c == 2)
 {
  printf("n is a Prime number");
  }
  else
 {
  printf("n is not a Prime number");
  }
  getch( );
}
```

OUTPUT;

4. **Write a program to display the following pattern upto N rows taking the value of N from the user :**

1
2 3
4 5 6
7 8 9 10

CODE:

```
#include<stdio.h>
#include<conio.h>
int main()
{
int n, i, j, c = 1;
printf("Enter the number of rows: ");
scanf("%d",&n);
for(i = 1; i <= n; i++)
{
for(j = 1; j <= i; ++j)
{
printf("%d ", c);
++c;
}
printf("\n");
}
getch( ):
}
```

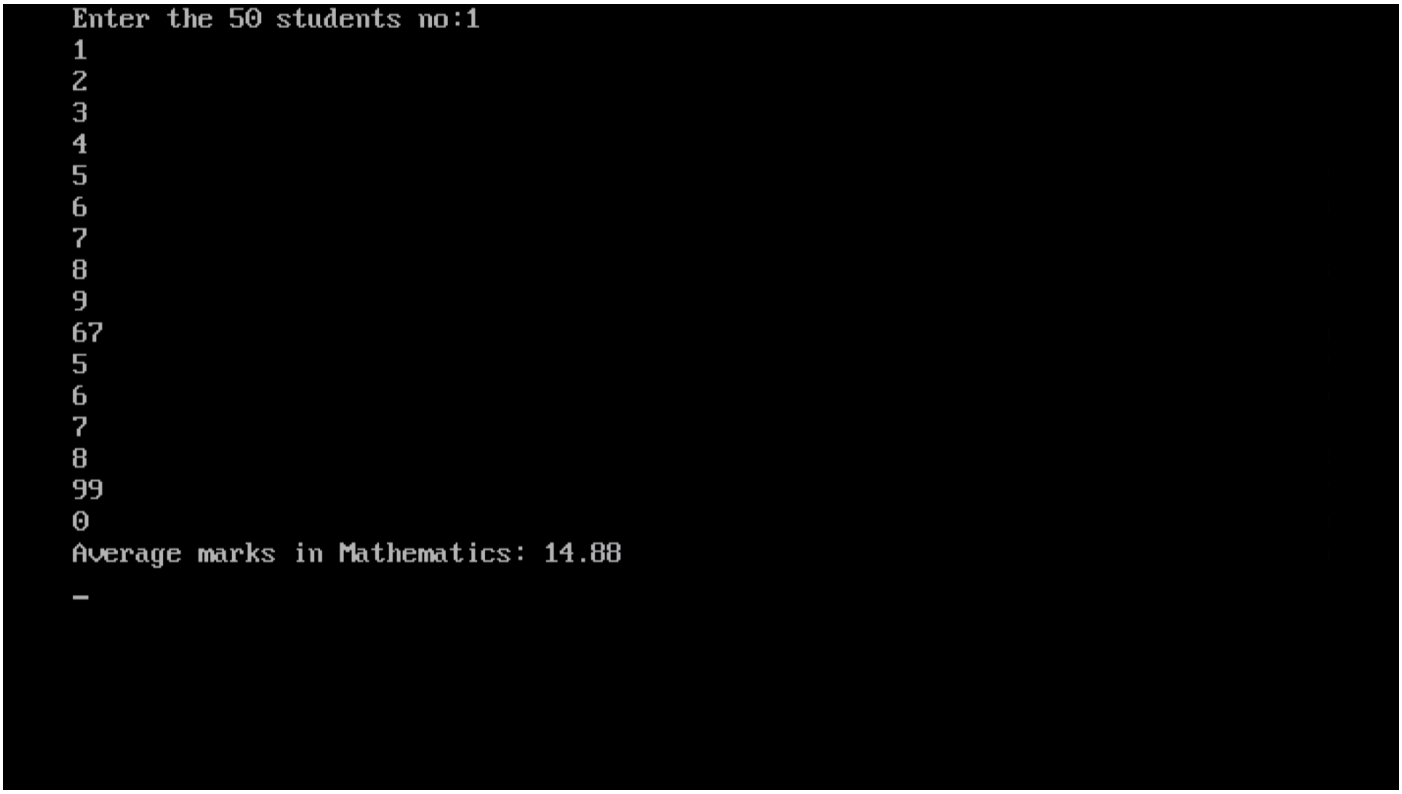
OUTPUT:

5. Write a program to input marks of 50 students using an array and display the average marks of the class.

CODE:

```
#include <stdio.h>
#include<conio.h>
int main()
{
        int marks[50], m, i, a=0, total=0;
        float f;
        printf("Enter the 50 students no: ");
        for(i = 0; ; i++)
        {
                scanf("%d", &marks[i]);
                if(marks[i] <= 0) {
                break;
                }
                a++;
                total += marks[i];
        }
        f = (float)total/(float)a;
        printf("Average marks in Mathematics: %.2f\n", f);
        getch( );
}
```

OUTPUT:

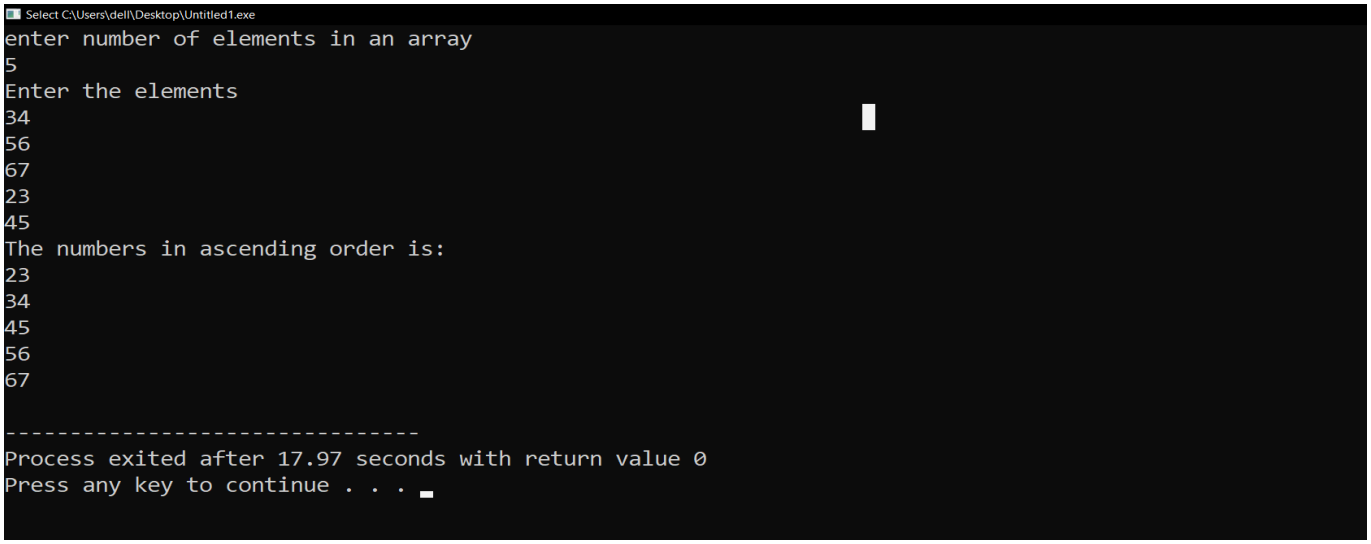


6. **Write a program to search for a number entered by the user in a given array and display the array in ascending order.**

CODE:

```
#include <stdio.h>
#include<conio.h>
int main ()
{
  int num[20];
  int i, j, a, n;
  printf("enter number of elements in an array\n");
  scanf("%d", &n);
  printf("Enter the elements\n");
  for (i = 0; i < n; ++i)
    scanf("%d", &num[i]);
  for (i = 0; i < n; ++i){
    for (j = i + 1; j < n; ++j){
      if (num[i] > num[j]){
        a = num[i];
        num[i] = num[j];
        num[j] = a;
      }
    }
  }
  printf("The numbers in ascending order is:\n");
  for (i = 0; i < n; ++i){
    printf("%d\n", num[i]);
  }
}
```

OUTPUT:

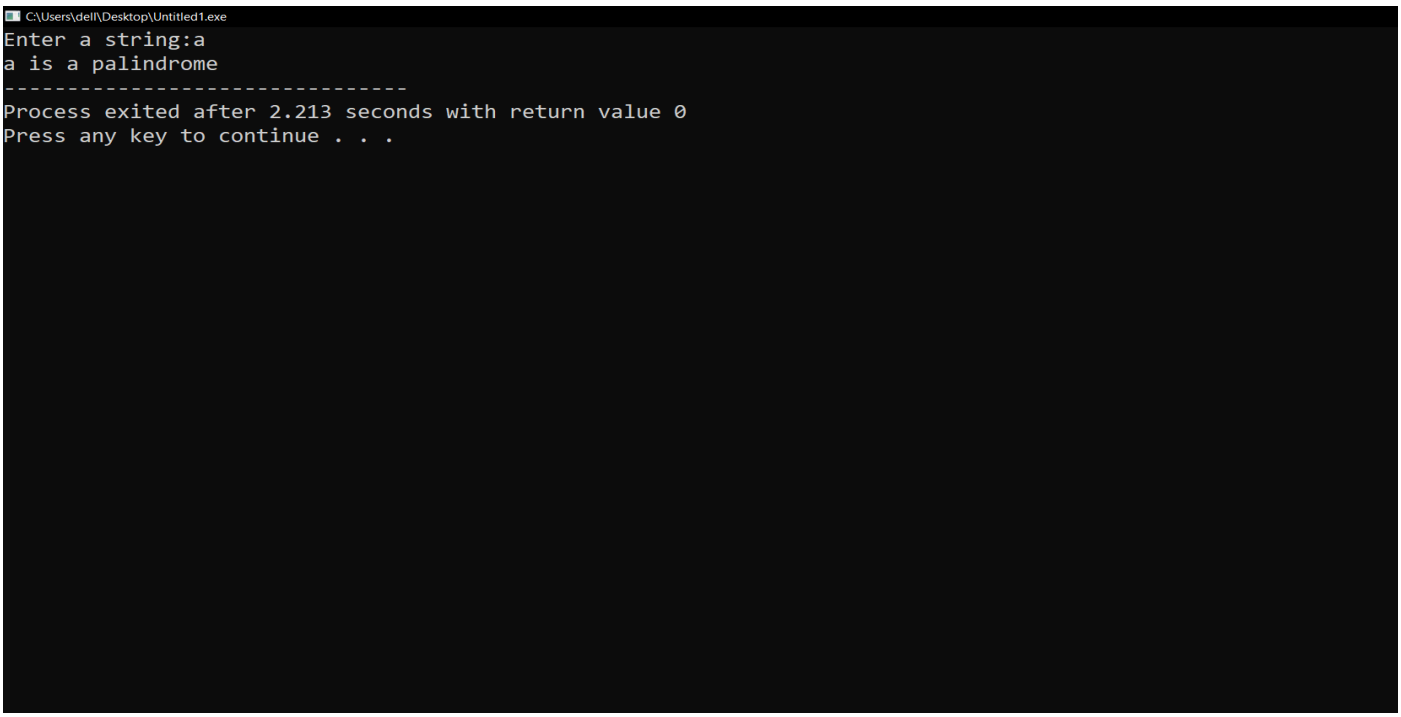


7. Write a program to check if a string is palindrome or not.

CODE:

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

int main()
{
   char string1[20];
   int i, length;
   int flag = 0;

   printf("Enter a string:");
   scanf("%s", string1);

   length = strlen(string1);

   for(i=0;i < length ;i++){
      if(string1[i] != string1[length-i-1]){
         flag = 1;
         break;
  }
}

   if (flag) {
      printf("%s is not a palindrome", string1);
   }
   else {
      printf("%s is a palindrome", string1);
   }
   return 0;
}
```

OUTPUT:

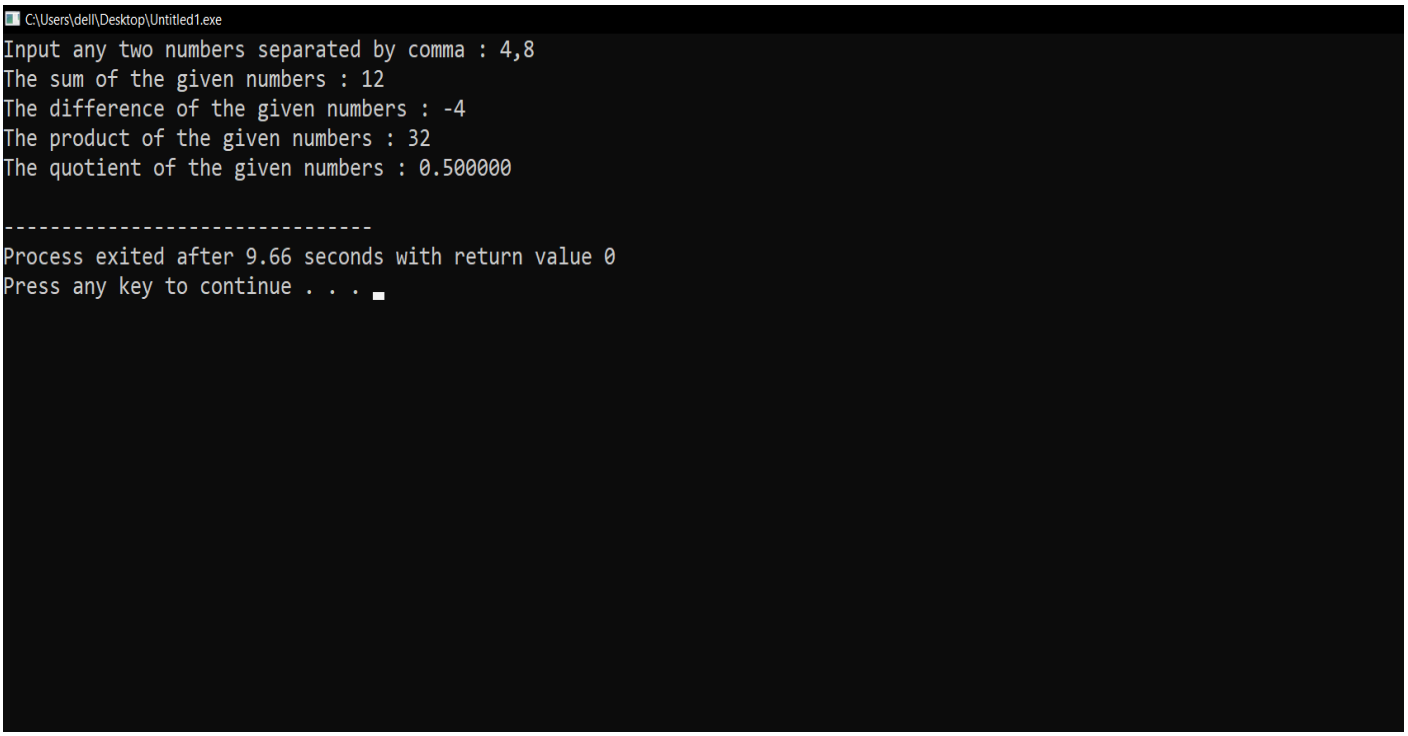


8. Write a program to add , subtract , multiply and divide two numbers using pointers.

CODE:

```
#include <stdio.h>
int main()
{
   int num1, num2;
   int sum, sub, mult, mod;
   float div;
   printf("Input any two numbers separated by comma : ");
   scanf("%d,%d", &num1, &num2);
   sum = num1 + num2;
   sub = num1 - num2;
   mult = num1 * num2;
   div = (float)num1 / num2;

   printf("The sum of the given numbers : %d\n", sum);
   printf("The difference of the given numbers : %d\n", sub);
   printf("The product of the given numbers : %d\n", mult);
   printf("The quotient of the given numbers : %f\n", div);

   return 0;
}
```

OUTPUT:



APPLICATION BASED PRACTICALS

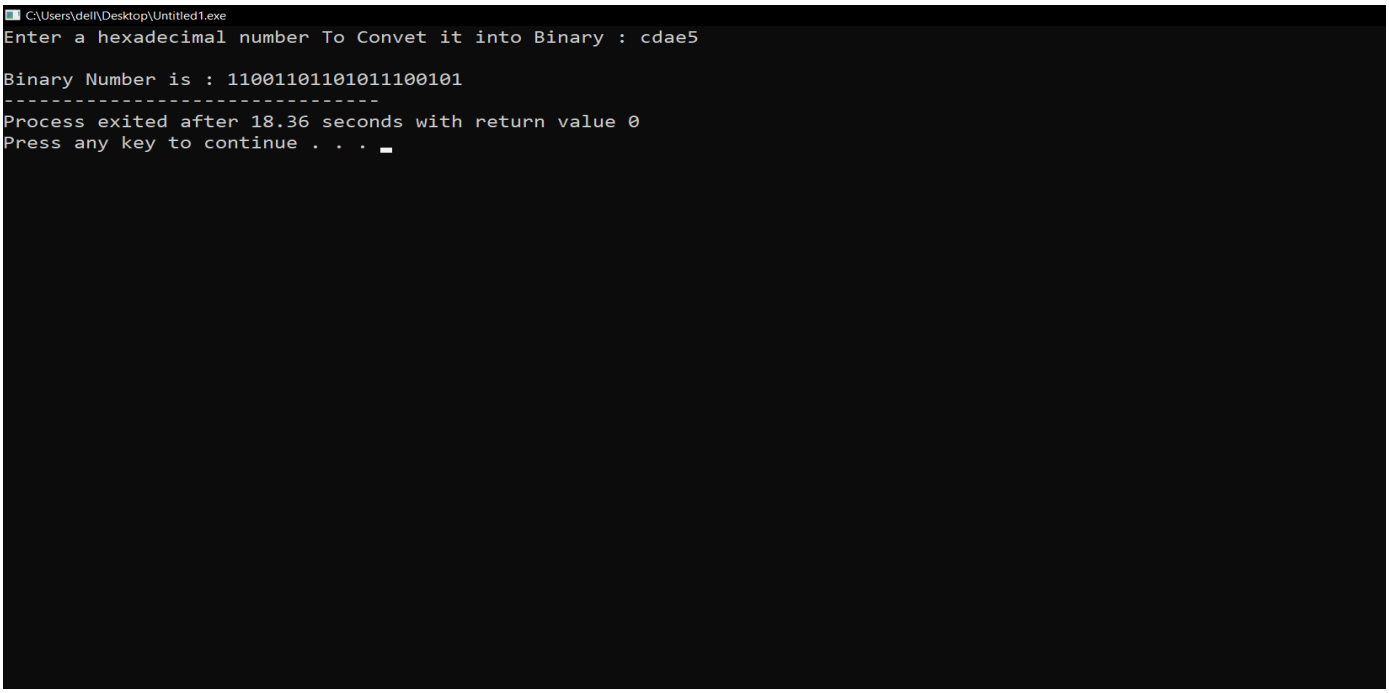
9. Write a program to convert a hexadecimal number into a binary number.

CODE:

```
#include<stdio.h>
int main()
{
	char hexNum[100];
	long int count=0;
	printf("Enter a hexadecimal number To Convet it into Binary : ");
	scanf("%s",hexNum);
	printf("\nBinary Number is : ");
	while(hexNum[count])
	{
	switch(hexNum[count])
	{
	case '0' : printf("0000");
	break;
	case '1' : printf("0001");
	break;
	case '2' : printf("0010");
	break;
	case '3' : printf("0011");
	break;
	case '4' : printf("0100");
	break;
	case '5' : printf("0101");
	break;
	case '6' : printf("0110");
	break;
	case '7' : printf("0111");
	break;
	case '8' : printf("1000");
	break;
	case '9' : printf("1001");
	break;
	case 'A' : printf("1010");
	break;
	case 'B' : printf("1011");
	break;
	case 'C' : printf("1100");
	break;
	case 'D' : printf("1101");
	break;
	case 'E' : printf("1110");
	break;
	case 'F' : printf("1111");
	break;
	case 'a' : printf("1010");
	break;
```

```
        case 'b' : printf("1011");
        break;
        case 'c' : printf("1100");
        break;
        case 'd' : printf("1101");
        break;
        case 'e' : printf("1110");
        break;
        case 'f' : printf("1111");
        break;
        default : printf("\nInvalid Entry, Please Try Again  %c",hexNum[count]);
        }
        count++;
        }
        return 0;
}
```

OUTPUT:

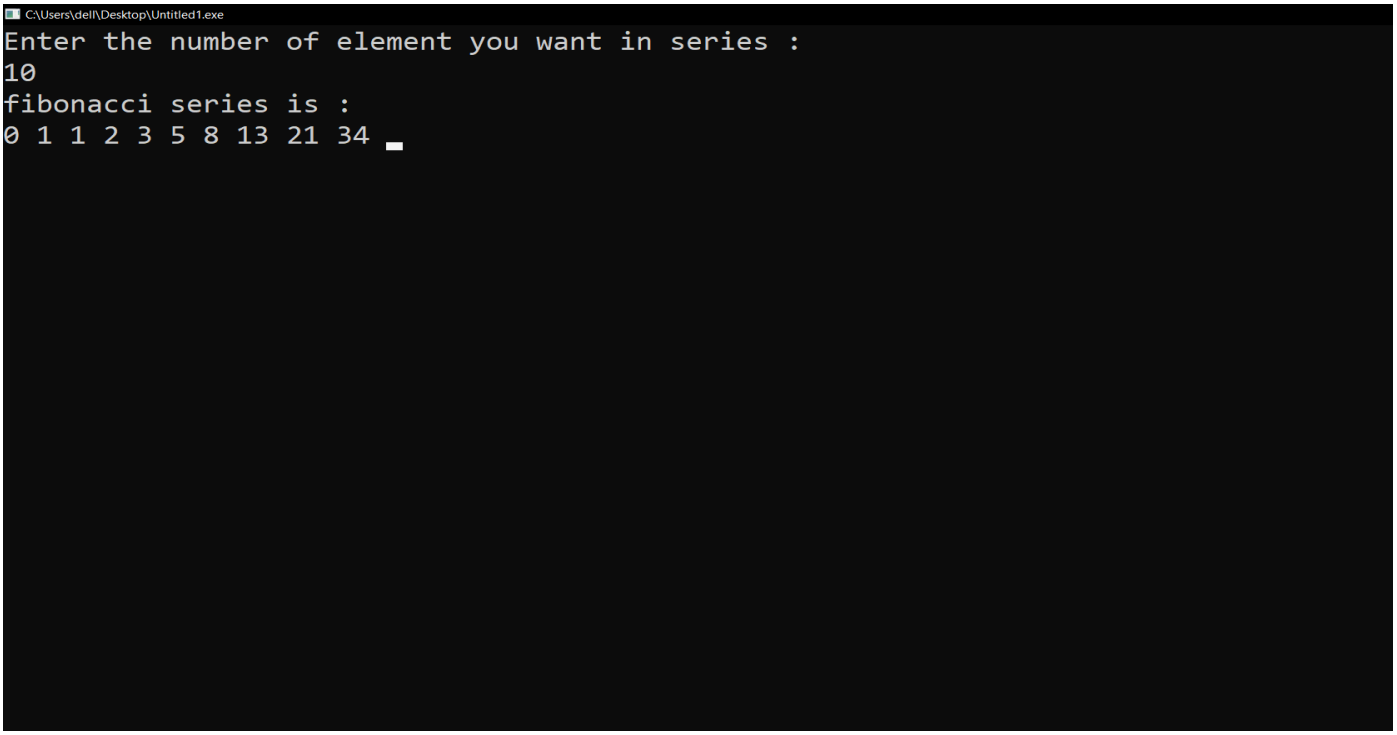


10. Write a program to calculate factorial of a number and display Fibonacci series upto n terms using recursive functions.

CODE:

```
#include<stdio.h>
#include<conio.h>
int fibonacci(int);
int main()
{
int n, i;
printf("Enter the number of element you want in series :\n");
scanf("%d",&n);
printf("fibonacci series is : \n");
for(i=0;i<n;i++) {
printf("%d ",fibonacci(i));
}
getch();
}

int fibonacci(int i){
if(i==0) return 0;
else if(i==1) return 1;
else return (fibonacci(i-1)+fibonacci(i-2));
}
```

OUTPUT:

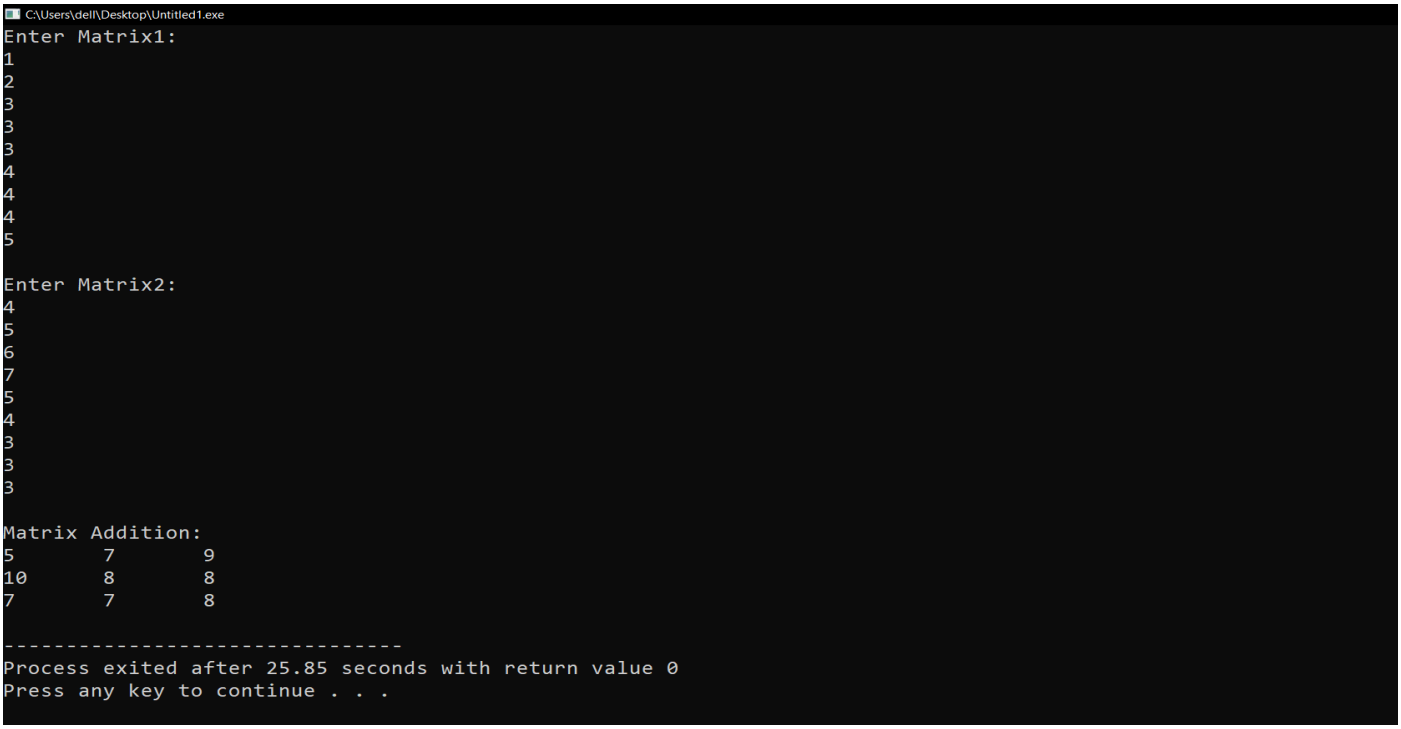


11. Write a program to perform
    (i) matrix addition ,
    (ii)matrix multiplication and
    (iii)matrix transpose on 2D array.

CODE:
(I).

```
#include<stdio.h>
#define COLUMN 3
#define ROW 3
int main()
{
int matrix1[ROW][COLUMN], matrix2[ROW][COLUMN], matrix3[ROW][COLUMN], matrix4[ROW][COLUMN];
int i,j;

printf("\nEnter Matrix1: \n");
for(i=0;i<ROW;i++)
{
for(j=0;j<COLUMN;j++)
{
scanf("%d", &matrix1[i][j]);
}
}
printf("\nEnter Matrix2: \n");
for(i=0;i<ROW;i++)
{
for(j=0;j<COLUMN;j++)
{
scanf("%d", &matrix2[i][j]);
}
}
for(i=0;i<ROW;i++)
{
for(j=0;j<COLUMN;j++)
{
matrix3[i][j] = matrix1[i][j] + matrix2[i][j];
}
}
printf("\nMatrix Addition: \n");
for(i=0;i<ROW;i++)
{
for(j=0;j<COLUMN;j++)
{
printf("%d\t", matrix3[i][j]);
}
printf("\n");
}
return 0;
}
```

OUTPUT:



(II).

```
#include <stdio.h>
void getMatrixElements(int matrix[][10], int row, int column)
{
printf("\nEnter elements: \n");
for (int i = 0; i < row; ++i)
 {
for (int j = 0; j < column; ++j)
{
printf("Enter a%d%d: ", i + 1, j + 1);
scanf("%d", &matrix[i][j]);
}
}
}
void multiplyMatrices(int first[][10],
int second[][10],
int result[][10],
int r1, int c1, int r2, int c2)
{
for (int i = 0; i < r1; ++i)
{
for (int j = 0; j < c2; ++j)
{
result[i][j] = 0;
}
}
for (int i = 0; i < r1; ++i)
{
for (int j = 0; j < c2; ++j)
{
for (int k = 0; k < c1; ++k)
{
```

```
result[i][j] += first[i][k] * second[k][j];
}
}
}
}
void display(int result[][10], int row, int column)
{
printf("\nOutput Matrix:\n");
for (int i = 0; i < row; ++i)
{
for (int j = 0; j < column; ++j)
{
printf("%d  ", result[i][j]);
if (j == column - 1)
printf("\n");
}
}
}
int main()
{
int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;
printf("Enter rows and column for the first matrix: ");
scanf("%d %d", &r1, &c1);
printf("Enter rows and column for the second matrix: ");
scanf("%d %d", &r2, &c2);
while (c1 != r2) {
printf("Error! Enter rows and columns again.\n");
printf("Enter rows and columns for the first matrix: ");
scanf("%d%d", &r1, &c1);
printf("Enter rows and columns for the second matrix: ");
scanf("%d%d", &r2, &c2);
}
getMatrixElements(first, r1, c1);
getMatrixElements(second, r2, c2);
multiplyMatrices(first, second, result, r1, c1, r2, c2);
display(result, r1, c2);
return 0;
}
```

OUTPUT:

```
C:\Users\dell\Desktop\Untitled1.exe
Enter elements:
Enter a11: 1
Enter a12: 2
Enter a13: 3
Enter a21: 4
Enter a22: 5
Enter a23: 6
Enter a31: 7
Enter a32: 8
Enter a33: 9

Enter elements:
Enter a11: 2
Enter a12: 3
Enter a13: 4
Enter a21: 5
Enter a22: 6
Enter a23: 7
Enter a31: 8
Enter a32: 9
Enter a33: 1

Output Matrix:
36  42  21
81  96  57
126  150  93

--------------------------------
Process exited after 44.6 seconds with return value 0
Press any key to continue . . .
```

(III).

```
#include <stdio.h>
int main()
{
int a[10][10], transpose[10][10], r, c;
printf("Enter rows and columns: ");
scanf("%d %d", &r, &c);
printf("\nEnter matrix elements:\n");
for (int i = 0; i < r; ++i)
for (int j = 0; j < c; ++j) {
printf("Enter element a%d%d: ", i + 1, j + 1);
scanf("%d", &a[i][j]);
}
printf("\nEntered matrix: \n");
for (int i = 0; i < r; ++i)
for (int j = 0; j < c; ++j) {
printf("%d  ", a[i][j]);
if (j == c - 1)
printf("\n");
}
for (int i = 0; i < r; ++i)
for (int j = 0; j < c; ++j) {
transpose[j][i] = a[i][j];
}
printf("\nTranspose of the matrix:\n");
for (int i = 0; i < c; ++i)
for (int j = 0; j < r; ++j) {
printf("%d  ", transpose[i][j]);
if (j == r - 1)
printf("\n");
}
return 0;
}
```

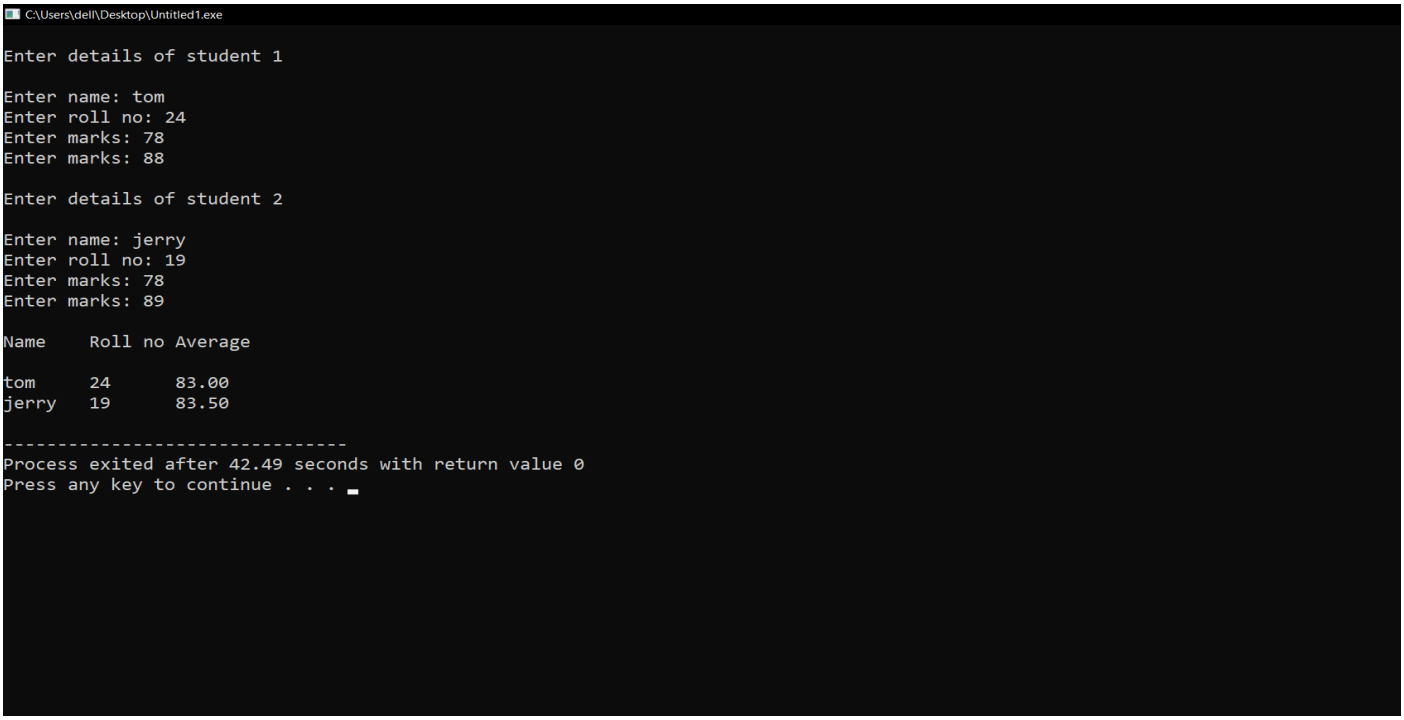
OUTPUT:

12. Write a program to make use of arrays with structure in the following ways:
    (i) Use as structured data member
    (ii) Create array of structure variables

CODE:
(I).

```
#include<string.h>
#define MAX 2
#define SUBJECTS 2
#include<stdio.h>

struct student
{
char name[20];
int roll_no;
float marks[SUBJECTS];
};
int main()
{
struct student arr_student[MAX];
int i, j;
float sum = 0;
for(i = 0; i < MAX; i++ )
{
printf("\nEnter details of student %d\n\n", i+1);

printf("Enter name: ");
scanf("%s", arr_student[i].name);

printf("Enter roll no: ");
scanf("%d", &arr_student[i].roll_no);

for(j = 0; j < SUBJECTS; j++)
{
printf("Enter marks: ");
scanf("%f", &arr_student[i].marks[j]);
}
}

printf("\n");

printf("Name\tRoll no\tAverage\n\n");

for(i = 0; i < MAX; i++ )
{
sum = 0;

for(j = 0; j < SUBJECTS; j++)
{
sum += arr_student[i].marks[j];
}
printf("%s\t%d\t%.2f\n",
arr_student[i].name, arr_student[i].roll_no, sum/SUBJECTS);
}
return 0;
```

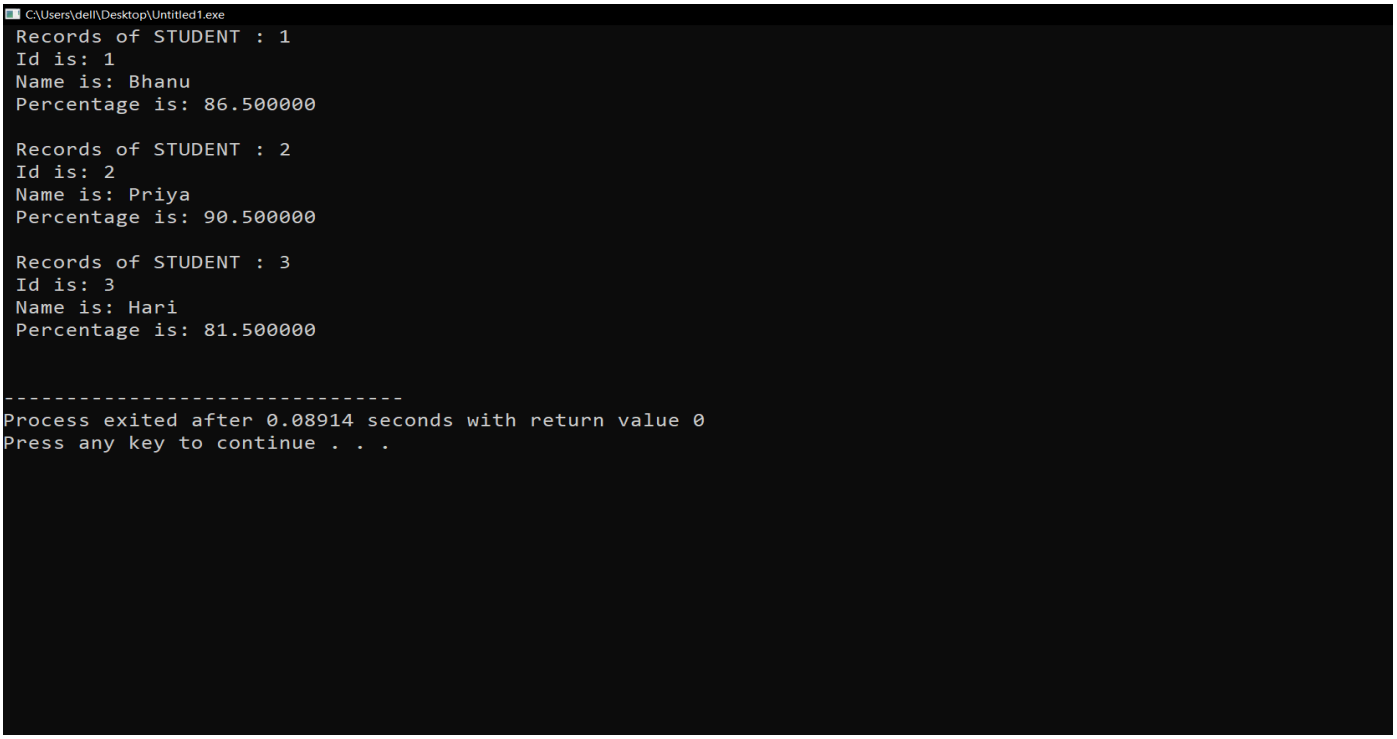
}
OUTPUT:



(II).

```
#include <stdio.h>
#include <string.h>
struct student
{
  int id;
  char name[30];
  float percentage;
};
int main()
{
  int i;
  struct student record[2];
  record[0].id=1;
  strcpy(record[0].name, "Bhanu");
  record[0].percentage = 86.5;
  record[1].id=2;
  strcpy(record[1].name, "Priya");
  record[1].percentage = 90.5;
  record[2].id=3;
  strcpy(record[2].name, "Hari");
  record[2].percentage = 81.5;
  for(i=0; i<3; i++){
    printf(" Records of STUDENT : %d \n", i+1);
    printf(" Id is: %d \n", record[i].id);
    printf(" Name is: %s \n", record[i].name);
    printf(" Percentage is: %f\n\n",record[i].percentage);
  }
  return 0;
}
```

OUTPUT:


C:\Users\dell\Desktop\Untitled1.exe
Records of STUDENT : 1
Id is: 1
Name is: Bhanu
Percentage is: 86.500000
Records of STUDENT : 2
Id is: 2
Name is: Priya
Percentage is: 90.500000
Records of STUDENT : 3
Id is: 3
Name is: Hari
Percentage is: 81.500000

Process exited after 0.08914 seconds with return value 0
Press any key to continue . . .

13. Write a program to read time in string format and extract hours minutes and seconds also check time validity.

CODE:

```
#include <stdio.h>
int ValidateTime(int hh , int mm , int ss)
{
   int ret=0;
   if(hh>24)   ret=1;
   if(mm>60)   ret=1;
   if(ss>60)   ret=1;
   return ret;
}
int main()
{
   char string[100]={0};
   int ret=0,hour=0,min=0,sec=0;
   printf("\nEnter the time in \"hh:mm:ss\" format : ");
   fgets(string,100,stdin);
   sscanf(string , "%d:%d:%d" , &hour,&min,&sec);
   ret = ValidateTime(hour,min,sec);
   if(ret)
   {
      printf("\nInvalid Time. Try Again.\n");
   }
   else
   {
      printf("\nThe Time is : %d:%d:%d\n",hour,min,sec);
   }
   return 0;
}
```

OUTPUT:

```
C:\Users\dell\Desktop\Untitled1.exe

Enter the time in "hh:mm:ss" format : 23:35:56

The Time is : 23:35:56

--------------------------------
Process exited after 12.99 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\dell\Desktop\Untitled1.exe

Enter the time in "hh:mm:ss" format : 25:45:45

Invalid Time. Try Again.

--------------------------------
Process exited after 7.713 seconds with return value 0
Press any key to continue . . .
```