

Programming for problem solving Lab

(MCAC191)

Assignment Submitted to

Prof. Surajit Dan (Asst. Professor)

Of

Maulana Abul Kalam Azad University of Technology, West Bengal

**MAULANA ABUL KALAM AZAD
UNIVERSITY OF TECHNOLOGY,
WEST BENGAL**



For

MCA

In

Department of Information Science

Submitted by:

Name : Suchorit Saha

Roll no :

Reg no :

Semester : I

/*

Write a program to calculate simple and compound interest

***/**

#include<stdio.h>

#include <math.h>

float simpleInterest(float principle,int roi,int time){

return ((principle*roi*time)/100);

}

double compoundInterest(float principle,int roi,int n,int time){

float r=(float)roi/100;

double a=principle*pow((1+(r/n)),n*time);

return a-principle;

}

int main(){

int c,r,t,n;

float p;

**printf("press 1 to calculate simple interest\npress 2 to
calculate compound interest\nyour choice:");**

scanf("%d",&c);

printf("Enter Priciple: ");

scanf("%f",&p);

printf("Enter Rate of Interest: ");

scanf("%d",&r);

printf("Enter Time: ");

scanf("%d",&t);

```

    if(c==2){
        printf("Enter Term: ");
        scanf("%d",&n);
        printf("%.2lf",compoundInterest(p,r,n,t));
    }
    else if(c==1){
        printf("%.2lf",simpleInterest(p,r,t));
    }
    return 0;
}

```

Output:

```

D:\Desktop\C(MCA)\1.exe
press 1 to calculate simple interest
press 2 to calculate compound interest
your choice:1
Enter Principle: 100
Enter Rate of Interest: 10
Enter Time: 1
10.00
-----
Process exited after 18.99 seconds with return value 0
Press any key to continue . . .

```

```

D:\Desktop\C(MCA)\1.exe
press 1 to calculate simple interest
press 2 to calculate compound interest
your choice:2
Enter Principle: 100
Enter Rate of Interest: 10
Enter Time: 1
Enter Term: 2
10.25
-----
Process exited after 10.7 seconds with return value 0
Press any key to continue . . .

```

```
/*
```

Write a program to swap values of two variables with and without using third variable

```
*/
```

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

```
void swap(int *p,int *q){
```

```
    *p=*p+*q;
```

```
    *q=*p-*q;
```

```
    *p=*p-*q;
```

```
    printf("Result 1st=%d & 2nd= %d",*p,*q);
```

```
}
```

```
int main(){
```

```
    int i,j;
```

```
    printf("Enter 1st number: ");
```

```
    scanf("%d",&i);
```

```
    printf("Enter 2nd number: ");
```

```
    scanf("%d",&j);
```

```
    printf("Result 1st=%d & 2nd= %d\n",i,j);
```

```
    swap(&i,&j);
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\2.exe  ×  +  ∨
Enter 1st number: 10
Enter 2nd number: 12
Before Swap 1st=10 & 2nd= 12
After Swap 1st=12 & 2nd= 10
-----
Process exited after 4.039 seconds with return value 0
Press any key to continue . . .
```

```
/*
```

Write a program to display the size of every data type using "sizeof" operator.

```
*/
```

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

```
int main() {
```

```
    printf("Size of char: %zu byte\n", sizeof(char));
```

```
    printf("Size of short: %zu bytes\n", sizeof(short));
```

```
    printf("Size of int: %zu bytes\n", sizeof(int));
```

```
    printf("Size of long: %zu bytes\n", sizeof(long));
```

```
    printf("Size of long long: %zu bytes\n", sizeof(long long));
```

```
    printf("Size of float: %zu bytes\n", sizeof(float));
```

```
    printf("Size of double: %zu bytes\n", sizeof(double));
```

```
    printf("Size of long double: %zu bytes\n", sizeof(long double));
```

```
    printf("Size of void*: %zu bytes\n", sizeof(void*));
```

```
    return 0;
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\3.exe  ×  +  v
Size of char: 1 byte
Size of short: 2 bytes
Size of int: 4 bytes
Size of long: 4 bytes
Size of long long: 8 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of long double: 16 bytes
Size of void*: 8 bytes

-----
Process exited after 0.0827 seconds with return value 0
Press any key to continue . . .
```

/*Write a program to illustrate the use of unary prefix and postfix increment and decrement operators.*/

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

```
int main(){
```

```
    int x;
```

```
    printf("Choose a digit: ");
```

```
    scanf("%d",&x);
```

```
    printf("Your digit is %d",x);
```

```
    printf("\nAfter post increment by 1 = %d",x++);//no change
```

```
    printf("\nAfter pre increment by 1 = %d",++x);//increase by
```

2

```
    printf("\nAfter post decrement by 1 = %d",x--);//no change
```

```
    printf("\nAfter pre decrement by 1 = %d",--x);//increase by
```

2

```
}
```

Output:

```
Choose a digit: 10
Your digit is 10
After post increment by 1 = 10
After pre increment by 1 = 12
After post decrement by 1 = 12
After pre decrement by 1 = 10
-----
Process exited after 5.281 seconds with return value 0
Press any key to continue . . .
```

```
/*
```

Write a program to input two numbers and display the maximum number.

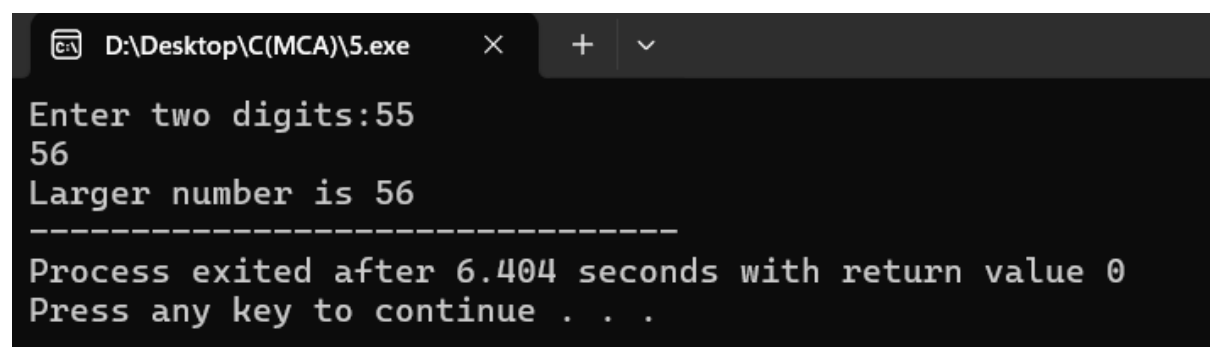
```
*/
```

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

```
int max(int i,int j){  
    if(i>j)  
        return i;  
    else  
        return j;  
}
```

```
int main(){  
    int i,j;  
    printf("Enter two digits:");  
    scanf("%d",&i);  
    scanf("%d",&j);  
    printf("Larger number is %d",max(i,j));  
}
```

Output:

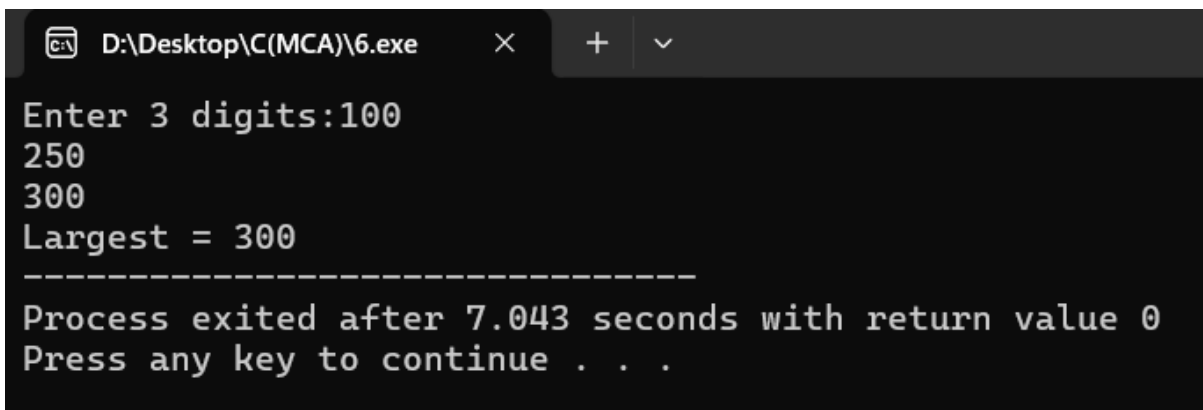


```
C:\> D:\Desktop\C(MCA)\5.exe  ×  +  ∨  
Enter two digits:55  
56  
Larger number is 56  
-----  
Process exited after 6.404 seconds with return value 0  
Press any key to continue . . .
```

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

```
int main(){  
    int a,b,c,l;  
  
    printf("Enter 3 digits:");  
    scanf("%d %d %d",&a,&b,&c);  
  
    l=(a>b)?((a>c)?a:c):((b>c)?b:c);  
    printf("Largest = %d",l);  
}
```

Output:



```
D:\Desktop\C(MCA)\6.exe  
Enter 3 digits:100  
250  
300  
Largest = 300  
-----  
Process exited after 7.043 seconds with return value 0  
Press any key to continue . . .
```

```
/*
```

Write a program to print number in reverse order with a difference of 2.

```
*/
```

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

```
void revPrint(int upperBound){  
    for(int i=upperBound;i>=0;i-=2){  
        printf("%d\t",i);
```



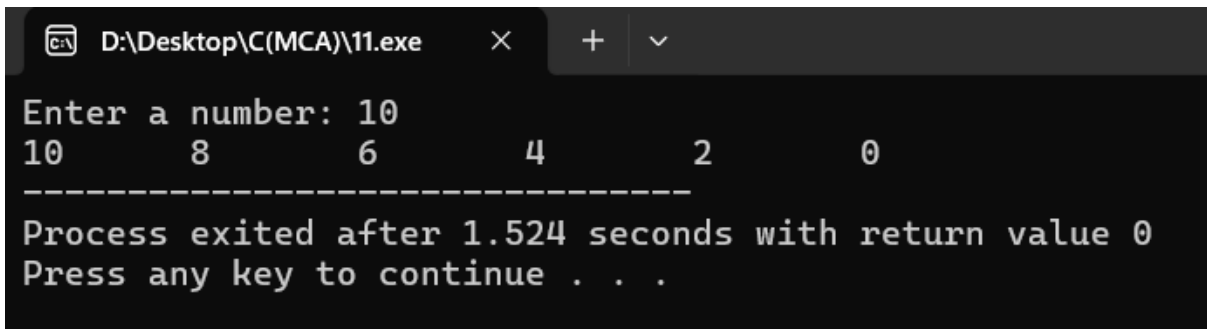
```

    }
}

int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    revPrint(n);
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\11.exe
Enter a number: 10
10      8      6      4      2      0
-----
Process exited after 1.524 seconds with return value 0
Press any key to continue . . .

```

```
/*
```

Write a program to count number of digits in a given integer

```
*/
```

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

```

int count(int num){
    int count=0;
    while(num>0){
        count++;
        num=num/10;
    }
}

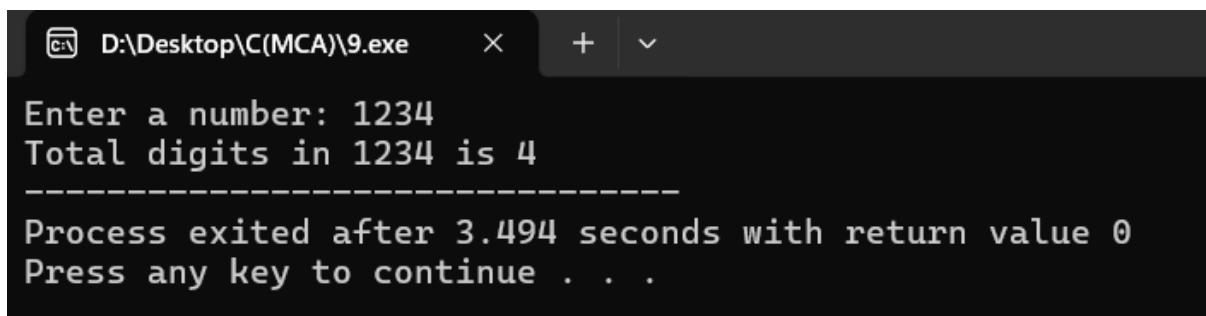
```

```

    }
    return count;
}
int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    printf("Total digits in %d is %d",n,count(n));
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\9.exe
Enter a number: 1234
Total digits in 1234 is 4
-----
Process exited after 3.494 seconds with return value 0
Press any key to continue . . .

```

/*

Write a program to reverse a given integer

***/**

#include<stdio.h>

```

int reverse(int num){
    int rev,temp;
    while(num>0){
        temp=num%10;
        rev=(rev*10)+temp;
        num/=10;
    }
}

```

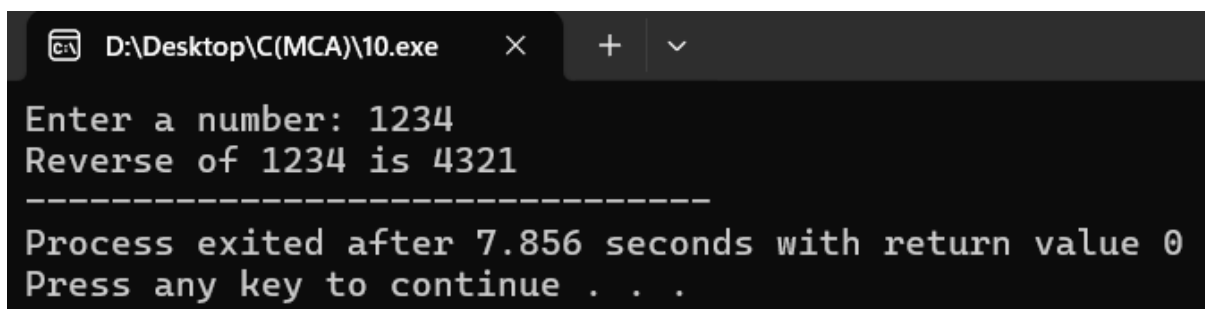
```

    }
    return rev;
}

int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    printf("Reverse of %d is %d",n,reverse(n));
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\10.exe
Enter a number: 1234
Reverse of 1234 is 4321
-----
Process exited after 7.856 seconds with return value 0
Press any key to continue . . .

```

/*Write a program to input name, marks of 5 subjects of a student and display the name of the student, the total marks scored, percentage scored and the class of result*/

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

```

struct Student {
    char name[50];
    int marks[5];
    int totalMarks;
    float percentage;

```

```
    char resultClass[20];
};

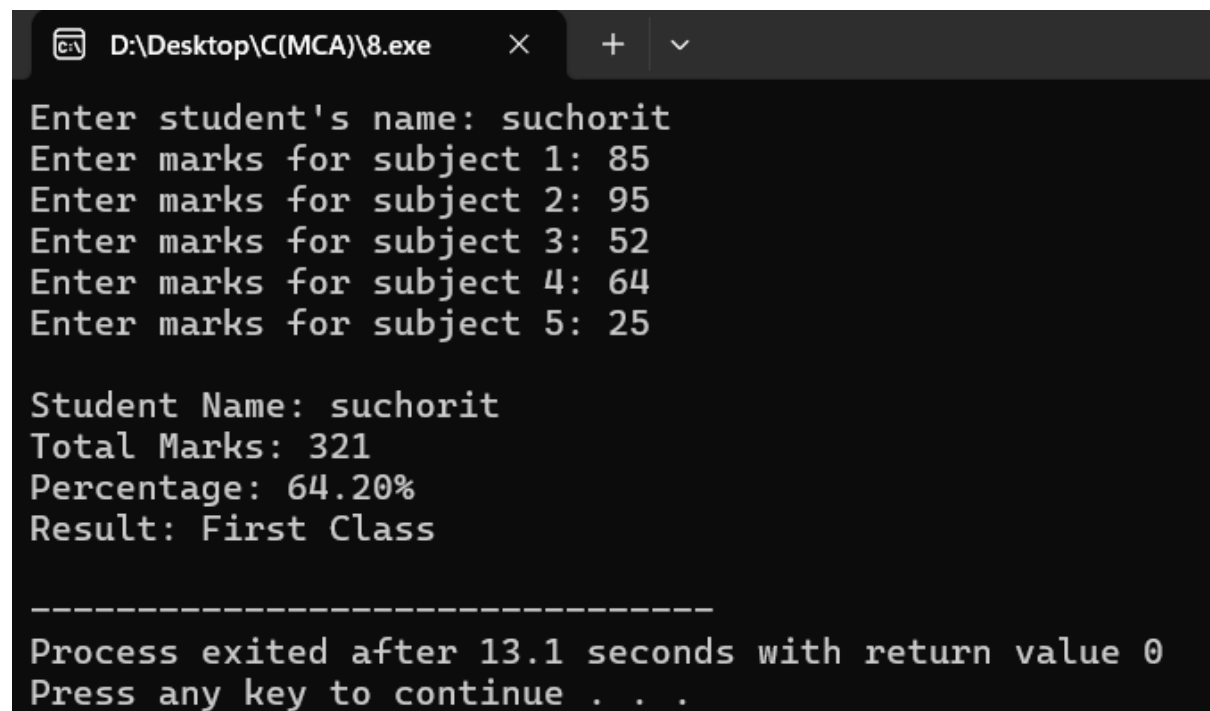
void calculateResult(struct Student *student) {
    student->totalMarks = 0;
    for (int i = 0; i < 5; i++) {
        student->totalMarks += student->marks[i];
    }
    student->percentage = (student->totalMarks / 5.0);
    if (student->percentage >= 85) {
        printf(student->resultClass, "Distinction");
    } else if (student->percentage >= 60) {
        printf(student->resultClass, "First Class");
    } else if (student->percentage >= 50) {
        printf(student->resultClass, "Second Class");
    } else if (student->percentage >= 35) {
        printf(student->resultClass, "Pass");
    } else {
        printf(student->resultClass, "Fail");
    }
}
```

```
int main() {
    struct Student student;
    printf("Enter student's name: ");
    scanf("%s", student.name);

    for (int i = 0; i < 5; i++) {
```

```
    printf("Enter marks for subject %d: ", i + 1);
    scanf("%d", &student.marks[i]);
}
// Calculate total marks, percentage, and result classification
calculateResult(&student);
// Display the result
printf("\nStudent Name: %s\n", student.name);
printf("Total Marks: %d\n", student.totalMarks);
printf("Percentage: %.2f%%\n", student.percentage);
printf("Result: %s\n", student.resultClass);
return 0;
}
```

Output:



```
D:\Desktop\C(MCA)\8.exe
Enter student's name: suchorit
Enter marks for subject 1: 85
Enter marks for subject 2: 95
Enter marks for subject 3: 52
Enter marks for subject 4: 64
Enter marks for subject 5: 25

Student Name: suchorit
Total Marks: 321
Percentage: 64.20%
Result: First Class

-----
Process exited after 13.1 seconds with return value 0
Press any key to continue . . .
```

```
/*
```

Write a program to print the sum of digits of a number using for loop.

```
*/
```

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

```
int main(){
```

```
    int n,total=0;
```

```
    printf("How many number you want to add: ");
```

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

```
    int temp=0;
```

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

```
        printf("Enter a number: ");
```

```
        scanf("%d",&temp);
```

```
        total+=temp;
```

```
    }
```

```
    printf("Total = %d",total);
```

```
    return 0;
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\12.exe  ×  +  ∨

How many number you want to add: 5
Enter a number: 10
Enter a number: 20
Enter a number: 30
Enter a number: 40
Enter a number: 50
Total = 150
-----
Process exited after 11.93 seconds with return value 0
Press any key to continue . . .
```

/*

Write a program to check whether a number is Palindrome or not.

***/**

#include<stdio.h>

int reverse(int num){

int rev,temp;

while(num>0){

temp=num%10;

rev=(rev*10)+temp;

num/=10;

}

return rev;

}

int main(){

int n,rev;

printf("Enter a number: ");

scanf("%d",&n);

rev=n;

if(n==reverse(rev))

printf("%d is palindrome",n);

else

printf("%d is not palindrome",n);

return 0;

}

Output:

```
D:\Desktop\C(MCA)\13.exe  ×  +  ∨
Enter a number: 2002
2002 is palindrome
-----
Process exited after 5.407 seconds with return value 0
Press any key to continue . . .
```

```
D:\Desktop\C(MCA)\13.exe  ×  +  ∨
Enter a number: 2003
2003 is not palindrome
-----
Process exited after 1.681 seconds with return value 0
Press any key to continue . . .
```

/*

Write a program to generate Fibonacci series

*/

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

```
int fibo(int num){
    if(num<=1){
        return num;
    }
    else{
        return fibo(num-1)+fibo(num-2);
    }
}
```

```
int main(){
    int n;
    printf("Enter number terms: ");
    scanf("%d",&n);
```



```

        for(int i=0;i<n;i++){
            printf("%d\t",fibo(i));
        }
        return 0;
    }
}

```

Output:

```

D:\Desktop\C(MCA)\14.exe
Enter number terms: 5
0      1      1      2      3
-----
Process exited after 3.438 seconds with return value 0
Press any key to continue . . .

```

/*

If a four-digit number is input through the keyboard, write a program to obtain the sum of the first and last digit of this number

*/

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

```

int sumFirstLast(int num){
    int temp,total=0;
    for(int i=0;i<4;i++){
        temp=num%10;
        if(i==0 | i==3){
            total+=temp;
        }
        num/=10;
    }
}

```

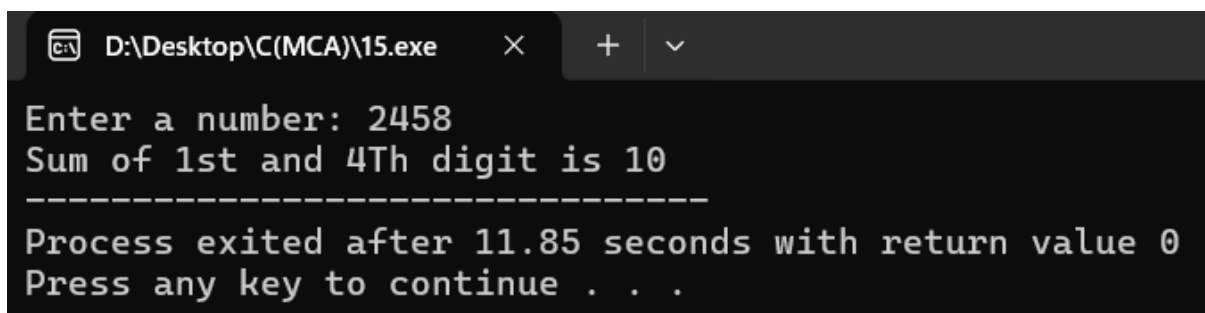
```

    }
    return total;
}

int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    printf("Sum of 1st and 4Th digit is %d",sumFirstLast(n));
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\15.exe
Enter a number: 2458
Sum of 1st and 4Th digit is 10
-----
Process exited after 11.85 seconds with return value 0
Press any key to continue . . .

```

/*

Write a program to find GCD (greatest common divisor or HCF) and LCM (least common multiple) of two numbers

***/**

```

#include<stdio.h>

int gcd(int a,int b){
    int m;
    m=a%b;
    while(m>0){

```

```

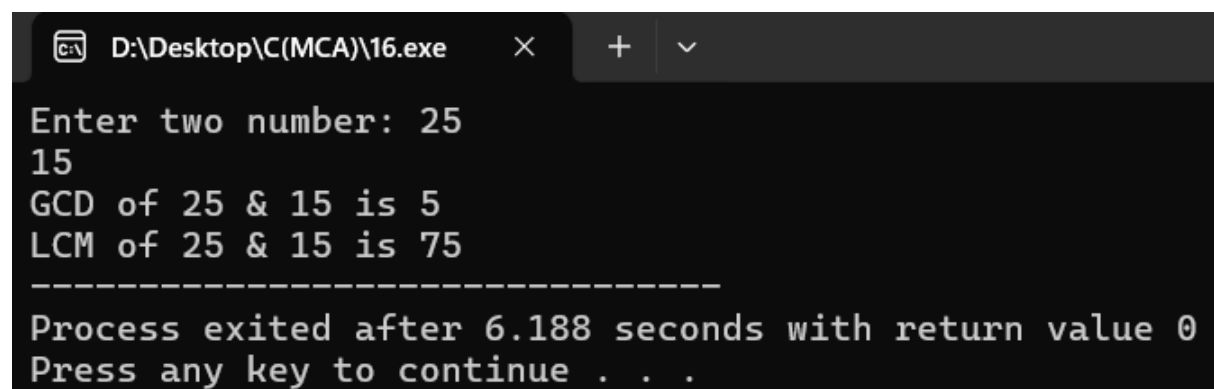
        a=b;
        b=m;
        m=a%b;
    }
    return b;
}

int lcm(int a,int b){
    return ((a*b)/gcd(a,b));
}

int main(){
    int n,m;
    printf("Enter two number: ");
    scanf("%d %d",&m,&n);
    printf("GCD of %d & %d is %d",m,n,gcd(m,n));
    printf("\nLCM of %d & %d is %d",m,n,lcm(m,n));
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\16.exe
Enter two number: 25
15
GCD of 25 & 15 is 5
LCM of 25 & 15 is 75
-----
Process exited after 6.188 seconds with return value 0
Press any key to continue . . .

```

```
/*
```

Program: Write a program to display the following pattern.

```
*
```

```
*
```

```
*
```

```
* *
```

```
* * * *
```

```
*/
```

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

```
int main(){
```

```
    for(int i=0;i<=5;i++){
```

```
        if(i<4){
```

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

```
        }
```

```
        else{
```

```
            for(int j=0;j<((i==4)?2:4);j++){
```

```
                printf("* ");
```

```
            }
```

```
            printf("\n");
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\17.exe
*
*
*
*
* *
* * * *

-----
Process exited after 0.1017 seconds with return value 0
Press any key to continue . . .
```

/*

Write a Program to find the largest and smallest element in Array

*/

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

```
int searchSmallest(int num[],int n){
```

```
    int s=num[0];
```

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

```
        if(num[i]<s){
```

```
            s=num[i];
```

```
        }
```

```
    }
```

```
    return s;
```

```
}
```

```
int main(){
```

```
    int num[]={42,87,16,93,56,24,67,35,78,12};
```

```
    int n=sizeof(num)/sizeof(num[0]);
```

```
    printf("Smallest element is %d",searchSmallest(num,n));
```

```
    return 0;
}
```

Output:

```
D:\Desktop\C(MCA)\18.exe × + ▾
Array:
42      87      16      93      56      24      67      35      78      12
Smallest element is 12
-----
Process exited after 0.09183 seconds with return value 0
Press any key to continue . . .
```

```
/*
```

Write a Program to reverse the array elements in C Programming.

```
*/
```

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

```
int main(){
```

```
    int num[]={42,87,16,93,56,24,67,35,78,12};
```

```
    int n=sizeof(num)/sizeof(num[0]);
```

```
    printf("\nActual array\n");
```

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

```
        printf("%d ",num[i]);
```

```
    }
```

```
    printf("\nReversed array\n");
```

```
    for(int i=n-1;i>=0;i--){
```

```
        printf("%d ",num[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\19.exe  X  +  v

Actual array
42 87 16 93 56 24 67 35 78 12
Reversed array
12 78 35 67 24 56 93 16 87 42
-----
Process exited after 0.1098 seconds with return value 0
Press any key to continue . . .
```

/*

Write a Program for deletion of an element from the specified location from Array

*/

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

```
int main(){
```

```
    int num[]={42,87,16,93,56,24,67,35,78,12};
```

```
    int n=sizeof(num)/sizeof(num[0]);
```

```
    printf("\nActual array\n");
```

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

```
        printf("%d ",num[i]);
```

```
    }
```

```
    int pos;
```

```
    printf("\nWhich position of array you wanna delete(array index start from 0): ");
```

```
    scanf("%d",&pos);
```

```

        for(int i=pos;i<=n;i++){
            num[i]=num[i+1];
        }

        printf("\nAfter Deletion at %d\n",pos);
        for(int i=0;i<n;i++){
            printf("%d ",num[i]);
        }
    }
}

```

Output:

```

D:\Desktop\C(MCA)\20.exe
Actual array
42 87 16 93 56 24 67 35 78 12
Which position of array you wanna delete(array index start from 0): 4

After Deletion at 4
42 87 16 93 24 67 35 78 12 25
-----
Process exited after 11.18 seconds with return value 0
Press any key to continue . . .

```

/*

Write a Program to access an element in 2-D Array.

*/

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

```
int main(){
```

```
    int row=3;
```

```
    int col=3;
```

```
    int nums[row][col]={
```

```
        {1,2,3},{4,5,6},{7,8,9}
```

```
    };
```



```

printf("2D Array:\n");
for(int i=0;i<row;i++){
    for(int j=0;j<col;j++){
        printf("%d ",nums[i][j]);
    }
    printf("\n");
}

printf("Select the row number and column number you
wanna access:\n");

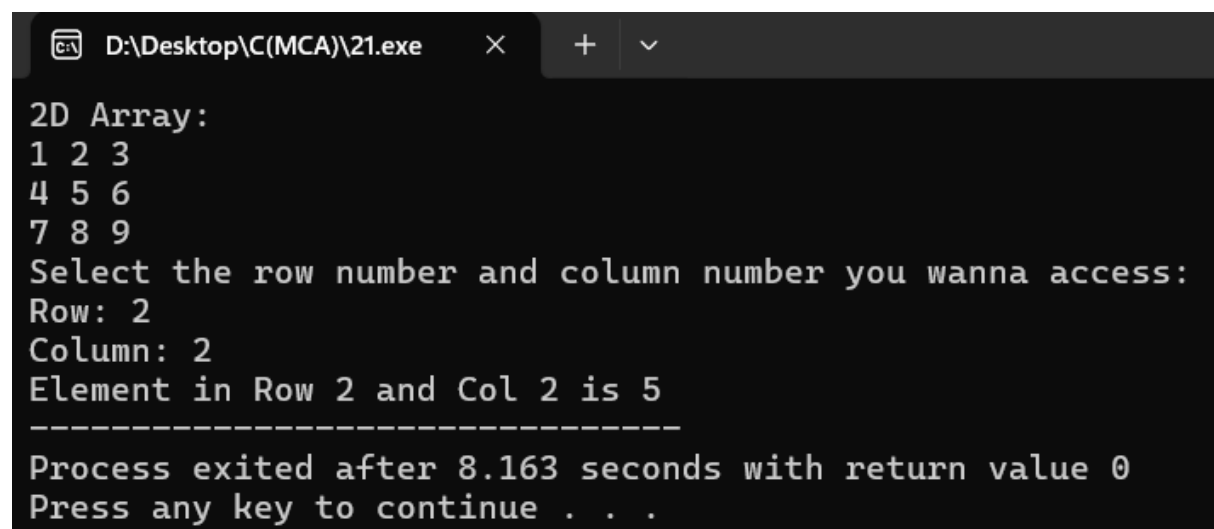
printf("Row: ");
scanf("%d",&row);
printf("Column: ");
scanf("%d",&col);

printf("Element in Row %d and Col %d is
%d",row,col,nums[row-1][col-1]);

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\21.exe
2D Array:
1 2 3
4 5 6
7 8 9
Select the row number and column number you wanna access:
Row: 2
Column: 2
Element in Row 2 and Col 2 is 5
-----
Process exited after 8.163 seconds with return value 0
Press any key to continue . . .

```

/*

Write a program for addition of two matrices of any order in C.

***/**

#include<stdio.h>

int main(){

int row=3;

int col=3;

int nums[row][col]={

{1,2,3},{4,5,6},{7,8,9}

};

int nums2[row][col]={

{1,2,3},{4,5,6},{7,8,9}

};

printf("1st 2D Array:\n");

for(int i=0;i<row;i++){

for(int j=0;j<col;j++){

printf("%d ",nums[i][j]);

}

printf("\n");

}

printf("\n2nd 2D Array:\n");

for(int i=0;i<row;i++){

for(int j=0;j<col;j++){

printf("%d ",nums2[i][j]);

}

printf("\n");

```

    }
    printf("\nSum of those 2D Array:\n");
    for(int i=0;i<row;i++){
        for(int j=0;j<col;j++){
            printf("%d ",nums[i][j]+nums2[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

Output:

```

D:\Desktop\C(MCA)\22.exe
1st 2D Array:
1 2 3
4 5 6
7 8 9

2nd 2D Array:
1 2 3
4 5 6
7 8 9

Sum of those 2D Array:
2 4 6
8 10 12
14 16 18

-----
Process exited after 0.09024 seconds with return value 0
Press any key to continue . . .

```

```
/*
```

```
Write a Program to multiply two 3 X 3 Matrices
```

```
*/
```

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

```
int main(){
```

```
    int row=3;
```

```
    int col=3;
```

```
    int nums[row][col]={
```

```
        {1,2,3},{4,5,6},{7,8,9}
```

```
    };
```

```
    int nums2[row][col]={
```

```
        {9,8,7},{6,5,4},{3,2,1}
```

```
    };
```

```
    int product[3][3];
```

```
    printf("1st 2D Array:\n");
```

```
    for(int i=0;i<row;i++){
```

```
        for(int j=0;j<col;j++){
```

```
            printf("%d ",nums[i][j]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```

printf("\n2nd 2D Array:\n");
for(int i=0;i<row;i++){
    for(int j=0;j<col;j++){
        printf("%d ",nums2[i][j]);
    }
    printf("\n");
}

for(int i=0;i<row;i++){
    for(int j=0;j<col;j++){
        product[i][j]=0;
        for(int k=0;k<row;k++){
            product[i][j]+=nums[i][k]*nums2[k][j];
        }
    }
}

printf("\nProduct of those 2D Array:\n");
for(int i=0;i<row;i++){
    for(int j=0;j<col;j++){
        printf("%d ",product[i][j]);
    }
    printf("\n");
}

return 0;

}

```

Output:

```
D:\Desktop\C(MCA)\23.exe  X  +  v

1st 2D Array:
1 2 3
4 5 6
7 8 9

2nd 2D Array:
9 8 7
6 5 4
3 2 1

Product of those 2D Array:
30 24 18
84 69 54
138 114 90

-----
Process exited after 0.08449 seconds with return value 0
Press any key to continue . . .
```

/*

Write a program to read a string and check for palindrome without using string related function (a string

is palindrome if its half is mirror by itself eg: abcdcba).

*/

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

```
void palindromeCheck(char s[],int n){
```

```
    int count=0;
```

```
    int i=0;
```

```
    int j=n-1;
```

```
    for(int i=0;(i<n/2);i++){
```

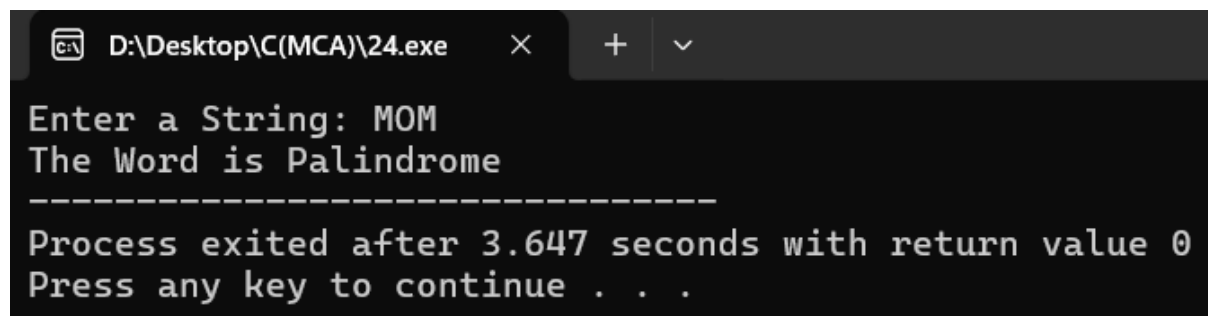
```
        if(s[i]!=s[n-1-i]){
```

```

        count++;
        break;
    }
}
if(count==0){
    printf("The Word is Palindrome");
}
else
    printf("Not a palindrome Word");
}
int main(){
    char s[100];
    int len=0;
    printf("Enter a String: ");
    scanf("%s",&s);
    while(s[len]!='\0'){
        len++;
    }
    palindromeCheck(s,len);
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\24.exe
Enter a String: MOM
The Word is Palindrome
-----
Process exited after 3.647 seconds with return value 0
Press any key to continue . . .

```

/*

Write a program to accept a string and count the number of vowels present in this String

***/**

#include<stdio.h>

int countVowel(char s[],int len){

int count=0;

for(int i=0;i<len;i++){

**if(s[i]=='A' || s[i]=='E' || s[i]=='I' || s[i]=='O' ||
s[i]=='U' || s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' ||
s[i]=='u')**

count++;

}

return count;

}

int main(){

char s[100];

int len=0;

printf("Enter a String: ");

scanf("%s",&s);

while(s[len]!='\0'){

len++;

}

printf("Number of vowel is %d",countVowel(s,len));

return 0;

}

Output:

```
D:\Desktop\C(MCA)\25.exe × + ∨
Enter a String: aeiou
Number of vowel is 5
-----
Process exited after 5.144 seconds with return value 0
Press any key to continue . . .
```

/*

Write a program to add, subtract, multiply and divide two integers using userdefined type function with

return type.

*/

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

```
int sum(int a,int b){
```

```
    return a+b;
```

```
}
```

```
int sub(int a,int b){
```

```
    return a-b;
```

```
}
```

```
int mul(int a,int b){
```

```
    return a*b;
```

```
}
```

```
int div(int a,int b){
```

```
    return a/b;
```

```
}
```

```
int main(){
```

```
    int a,b,c;
```

```
    printf("Enter 2 numbers: \n");
```

```
    scanf("%d %d",&a,&b);
```

```
printf("\npres 1 to add\npres 2 to subtract\npres 3 to
multiply\npres 4 to divide\nYour choice: ");
scanf("%d",&c);
switch(c){
    case 1:
        printf("Sum = %d",sum(a,b));
        break;
    case 2:
        printf("Differnce = %d",sub(a,b));
        break;
    case 3:
        printf("Product = %d",mul(a,b));
        break;
    case 4:
        printf("Qutaint = %d",div(a,b));
        break;
    default:
        printf("Wrong input!!!");
        break;
}
return 0;
}
```

Output:

```
D:\Desktop\C(MCA)\26.exe  ×  +  ∨

Enter 2 numbers:
12
15

press 1 to add
press 2 to subtract
press 3 to multiply
press 4 to divide
Your choice: 3
Product = 180
-----
Process exited after 9.229 seconds with return value 0
Press any key to continue . . .
```

/*

Write a program to calculate sum of first 20 natural numbers using recursive function

*/

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

```
int sum(int limit){
```

```
    if(limit<=1){
```

```
        return 1;
```

```
    }
```

```
    else{
```

```
        return limit+sum(limit-1);
```

```
    }
```

```
}
```

```
int main(){
```

```
    printf("Sum of first 20 natural number is %d",sum(20));
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\27.exe  X  +  v
Sum of first 20 natural number is 210
-----
Process exited after 0.09832 seconds with return value 0
Press any key to continue . . .
```

/*

Write a program to generate Fibonacci series using recursive function

*/

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

```
int fibo(int num){
```

```
    if(num<=1){
```

```
        return num;
```

```
    }
```

```
    else{
```

```
        return fibo(num-1)+fibo(num-2);
```

```
    }
```

```
}
```

```
int main(){
```

```
    int n;
```

```
    printf("Enter number terms: ");
```

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

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

```
        printf("%d\t",fibo(i));
```

```
    }
```

```
    return 0;
```

```
}
```

Output:

```
D:\Desktop\C(MCA)\28.exe  ×  +  ▾
Enter number terms: 5
0      1      1      2      3
-----
Process exited after 1.594 seconds with return value 0
Press any key to continue . . .
```

```
/*
```

Write a program to swap two integers using call by value and call by reference methods of passing

arguments to a function.

```
*/
```

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

```
void callByReference(int *p,int *q){
```

```
    *p=*p+*q;
```

```
    *q=*p-*q;
```

```
    *p=*p-*q;
```

```
    printf("After call by reference: %d\t%d",*p,*q);
```

```
}
```

```
void callByValue(int p,int q){
```

```
    p=p+q;
```

```
    q=p-q;
```

```
    p=p-q;
```

```
    printf("After call by reference: %d\t%d",p,q);
```

```
}
```

```

int main(){
    int m,n;

    printf("Enter 1st number: ");
    scanf("%d",&m);

    printf("Enter 2nd number: ");
    scanf("%d",&n);

    callByValue(m,n);

    printf("\nValue of 1st and 2nd number after call by value %d
and %d (Remain unchanged in memory)\n",m,n);

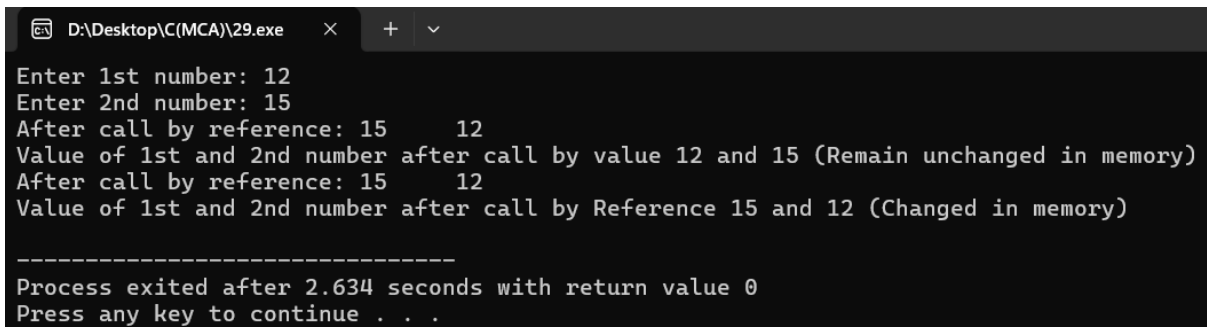
    callByReference(&m,&n);

    printf("\nValue of 1st and 2nd number after call by Reference
%d and %d (Changed in memory)\n",m,n);

    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\29.exe
Enter 1st number: 12
Enter 2nd number: 15
After call by reference: 15      12
Value of 1st and 2nd number after call by value 12 and 15 (Remain unchanged in memory)
After call by reference: 15      12
Value of 1st and 2nd number after call by Reference 15 and 12 (Changed in memory)

-----
Process exited after 2.634 seconds with return value 0
Press any key to continue . . .

```

```
/*
```

Write a program to find sum of digits of the number using Recursive Function

```
*/
```

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

```
int sum(int num){
```

```

        static int total=0;
        if(num>0){
            total+=num%10;
            return sum(num/10);
        }

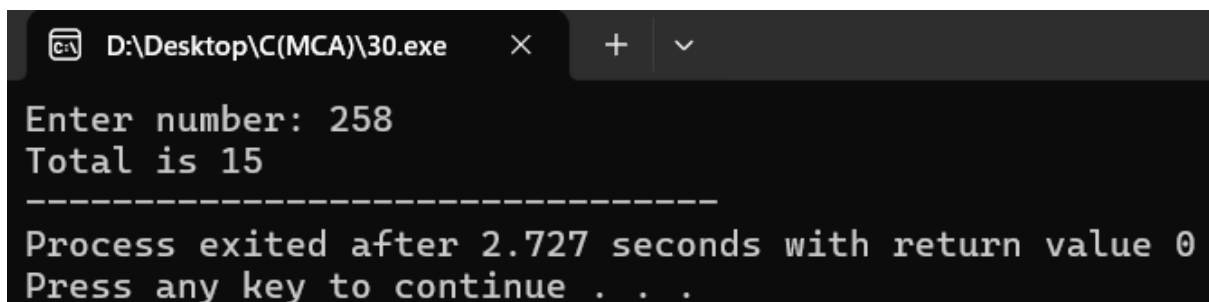
        return total;
    }

int main(){
    int n;
    printf("Enter number: ");
    scanf("%d",&n);

    printf("Total is %d",sum(n));
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\30.exe
Enter number: 258
Total is 15
-----
Process exited after 2.727 seconds with return value 0
Press any key to continue . . .

```

/*

Write a program to read an integer number and print the reverse of that number using recursion

***/**

#include<stdio.h>

```

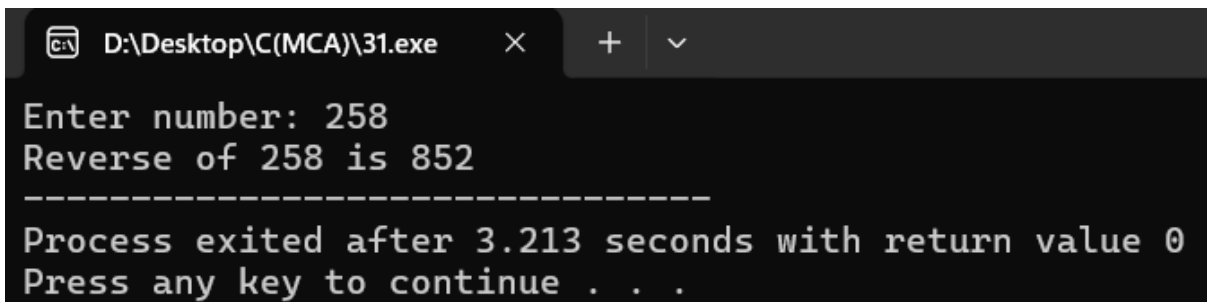
int revRec(int num){
    static int rev=0;
    if(num>0){
        rev=(rev*10)+(num%10);
        revRec(num/10);
    }
    return rev;
}

int main(){
    int n;
    printf("Enter number: ");
    scanf("%d",&n);

    printf("Reverse of %d is %d",n,revRec(n));
    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\31.exe
Enter number: 258
Reverse of 258 is 852
-----
Process exited after 3.213 seconds with return value 0
Press any key to continue . . .

```

/*

Write a C program to find maximum and minimum between two numbers using functions.

***/**

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



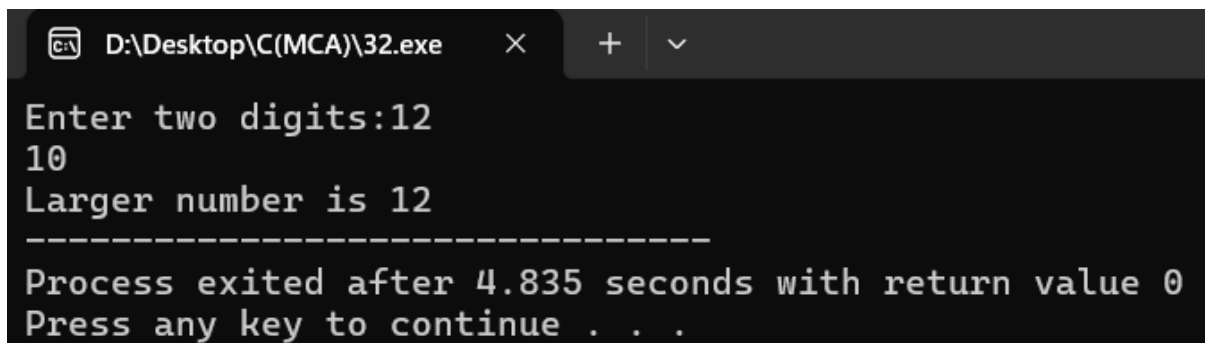
```

int max(int i,int j){
    if(i>j)
        return i;
    else
        return j;
}

int main(){
    int i,j;
    printf("Enter two digits:");
    scanf("%d",&i);
    scanf("%d",&j);
    printf("Larger number is %d",max(i,j));
}

```

Output:



```

D:\Desktop\C(MCA)\32.exe
Enter two digits:12
10
Larger number is 12
-----
Process exited after 4.835 seconds with return value 0
Press any key to continue . . .

```

/*

Write a C program to check whether a number is even or odd using functions.

***/**

#include<stdio.h>

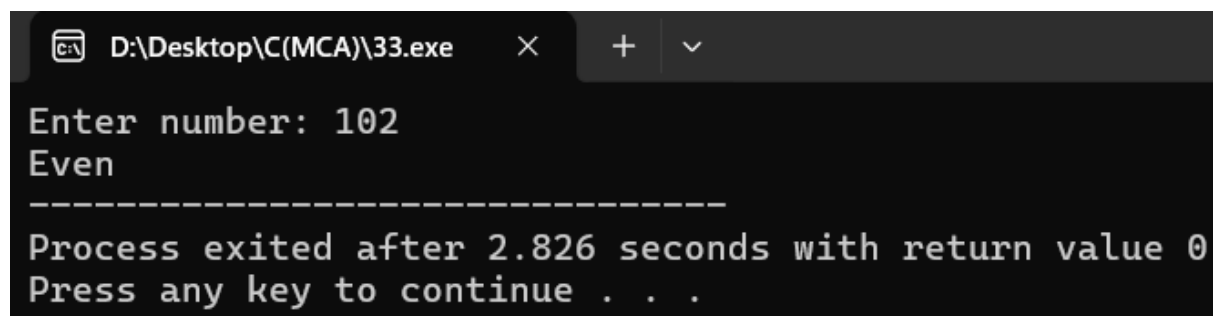
```
void oddEven(int num){
    if(num%2==0)
        printf("Even");
    else
        printf("Odd");
}

int main(){
    int n;

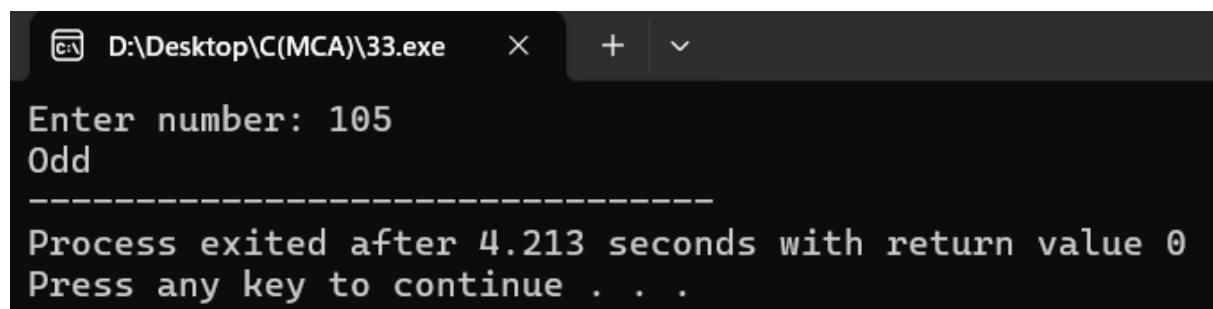
    printf("Enter number: ");
    scanf("%d",&n);

    oddEven(n);
}
```

Output:



```
D:\Desktop\C(MCA)\33.exe  ×  +  ▾
Enter number: 102
Even
-----
Process exited after 2.826 seconds with return value 0
Press any key to continue . . .
```



```
D:\Desktop\C(MCA)\33.exe  ×  +  ▾
Enter number: 105
Odd
-----
Process exited after 4.213 seconds with return value 0
Press any key to continue . . .
```

/*

Write a C program to check whether a number is prime, Armstrong or perfect number using functions

***/**

#include <stdio.h>

#include <math.h>

// Function to check if a number is prime

int isPrime(int num) {

if (num <= 1) return 0; // Not prime

for (int i = 2; i <= sqrt(num); i++) {

if (num % i == 0) return 0; // Divisible, not prime

}

return 1; // Prime

}

// Function to check if a number is an Armstrong number

int isArmstrong(int num) {

int originalNum = num, sum = 0, digits = 0;

// Count the number of digits

while (originalNum != 0) {

digits++;

originalNum /= 10;

}

originalNum = num;

// Calculate the sum of the power of digits

while (originalNum != 0) {

```
    int digit = originalNum % 10;
    sum += pow(digit, digits);
    originalNum /= 10;
}

return sum == num; // Return true if sum equals the original
number
}
```

// Function to check if a number is a perfect number

```
int isPerfect(int num) {
    int sum = 0;
    for (int i = 1; i <= num / 2; i++) {
        if (num % i == 0) sum += i; // Add divisors
    }
    return sum == num; // Perfect number if sum equals the number
}
```

// Main function

```
int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    // Check if the number is prime
    if (isPrime(num)) {
        printf("%d is a Prime number.\n", num);
    } else {
```

```

    printf("%d is not a Prime number.\n", num);
}

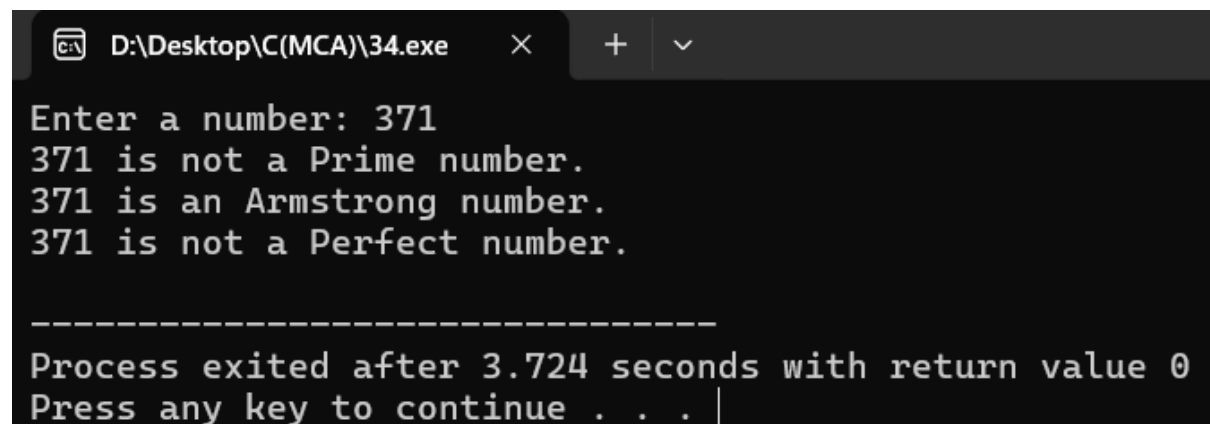
// Check if the number is an Armstrong number
if (isArmstrong(num)) {
    printf("%d is an Armstrong number.\n", num);
} else {
    printf("%d is not an Armstrong number.\n", num);
}

// Check if the number is a perfect number
if (isPerfect(num)) {
    printf("%d is a Perfect number.\n", num);
} else {
    printf("%d is not a Perfect number.\n", num);
}

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\34.exe  X  +  v
Enter a number: 371
371 is not a Prime number.
371 is an Armstrong number.
371 is not a Perfect number.

-----
Process exited after 3.724 seconds with return value 0
Press any key to continue . . . |

```

/*

Write a C program to find power of any number using recursion

***/**

#include <stdio.h>

// Function to calculate power using recursion

double power(double base, int exponent) {

if (exponent == 0) {

return 1; // Base case: any number raised to 0 is 1

} else if (exponent > 0) {

return base * power(base, exponent - 1); // Recursive case for positive exponent

} else {

return 1 / power(base, -exponent); // Handle negative exponent

}

}

// Main function

int main() {

double base;

int exponent;

printf("Enter the base: ");

scanf("%lf", &base);

printf("Enter the exponent: ");

```

scanf("%d", &exponent);

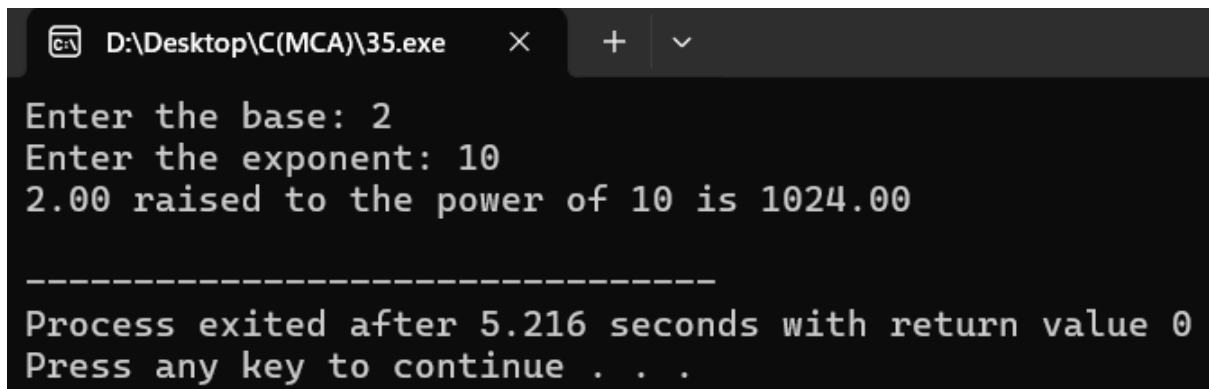
double result = power(base, exponent);

printf("%.2lf raised to the power of %d is %.2lf\n", base,
exponent, result);

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\35.exe
Enter the base: 2
Enter the exponent: 10
2.00 raised to the power of 10 is 1024.00

-----
Process exited after 5.216 seconds with return value 0
Press any key to continue . . .

```

// Question : Write a program to find the sum of all the elements of an array using pointers.

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

```

int main() {
    int n, i;
    int sum = 0;
    int *ptr;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

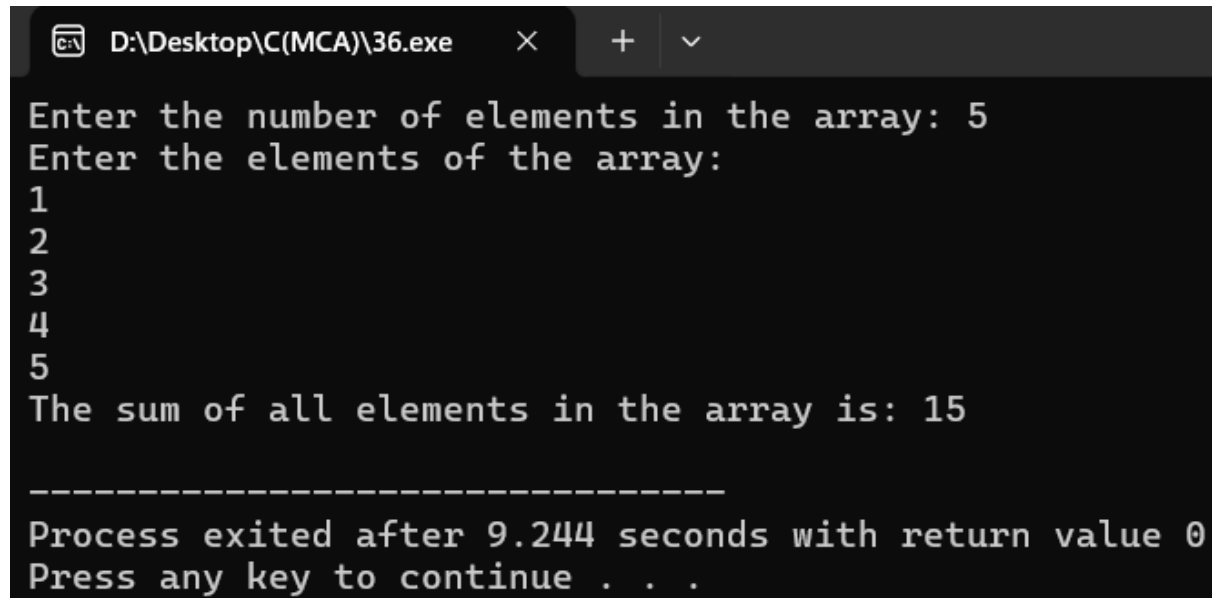
```

```

int arr[n];
printf("Enter the elements of the array:\n");
for (i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
}
ptr = arr;
for (i = 0; i < n; i++) {
    sum += *ptr;
    ptr++;
}
printf("The sum of all elements in the array is: %d\n", sum);
return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\36.exe
Enter the number of elements in the array: 5
Enter the elements of the array:
1
2
3
4
5
The sum of all elements in the array is: 15

-----
Process exited after 9.244 seconds with return value 0
Press any key to continue . . .

```

//Question: Write a program to swap value of two variables using pointer.

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



```
void swap(int *a, int *b) {  
    int temp;  
    temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
int main() {  
    int x, y;  
  
    printf("Enter the value of x: ");  
    scanf("%d", &x);  
    printf("Enter the value of y: ");  
    scanf("%d", &y);  
  
    printf("\nBefore swapping:\n");  
    printf("x = %d, y = %d\n", x, y);  
  
    swap(&x, &y);  
  
    printf("\nAfter swapping:\n");  
    printf("x = %d, y = %d\n", x, y);  
  
    return 0;  
}
```

Output:

```
D:\Desktop\C(MCA)\37.exe  X  +  v
Enter the value of x: 5
Enter the value of y: 6

Before swapping:
x = 5, y = 6

After swapping:
x = 6, y = 5

-----
Process exited after 6.655 seconds with return value 0
Press any key to continue . . .
```

//Question : Write a program to add two numbers using pointers.

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

```
int main() {
```

```
    int num1, num2, sum;
```

```
    int *ptr1, *ptr2;
```

```
    printf("Enter the first number: ");
```

```
    scanf("%d", &num1);
```

```
    printf("Enter the second number: ");
```

```
    scanf("%d", &num2);
```

```
    ptr1 = &num1;
```

```
ptr2 = &num2;
```

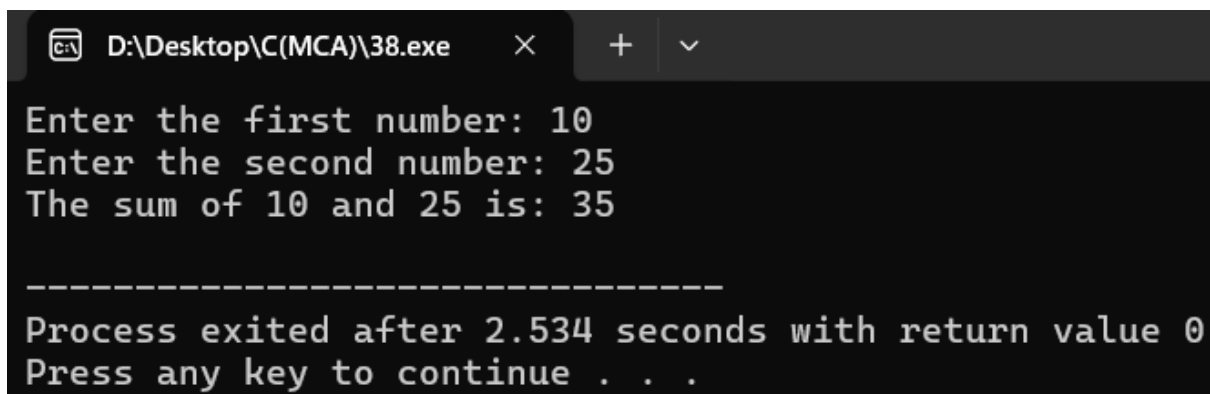
```
sum = *ptr1 + *ptr2;
```

```
printf("The sum of %d and %d is: %d\n", num1, num2, sum);
```

```
return 0;
```

```
}
```

Output:



```
Enter the first number: 10
Enter the second number: 25
The sum of 10 and 25 is: 35

-----
Process exited after 2.534 seconds with return value 0
Press any key to continue . . .
```

//Question : Write a program to input and print array elements using pointer.

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

```
int main() {
```

```
    int n, i;
```

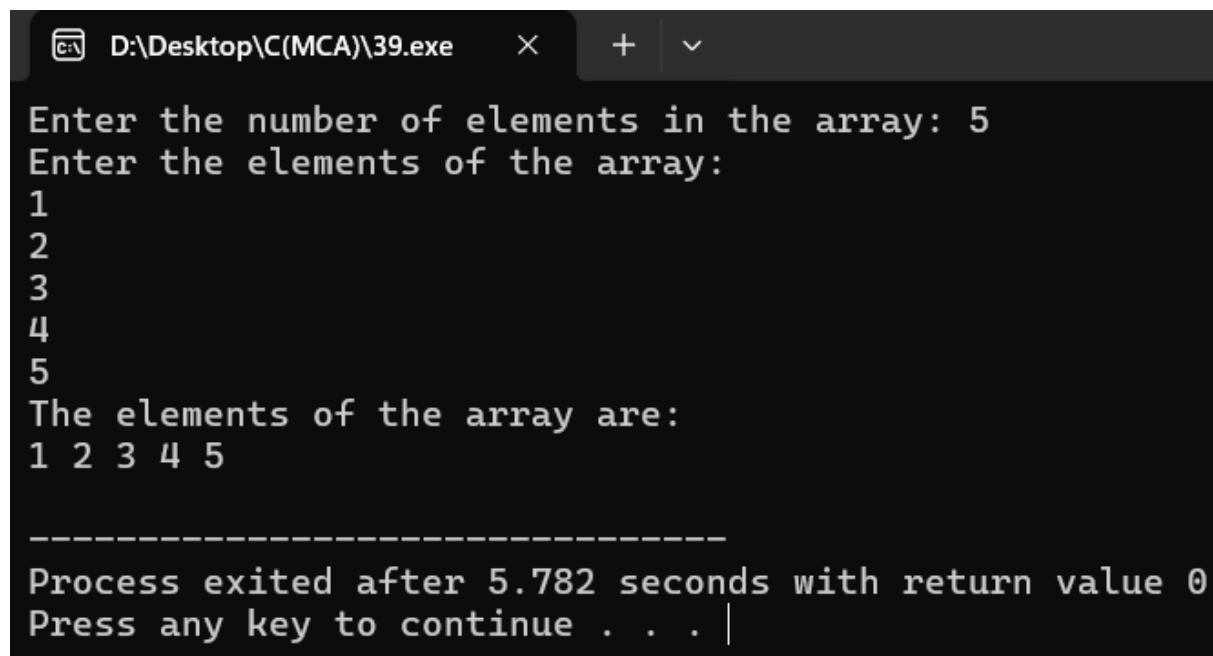
```
    int *ptr;
```

```
    printf("Enter the number of elements in the array: ");
```

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

```
int arr[n];  
ptr = arr;  
printf("Enter the elements of the array:\n");  
for (i = 0; i < n; i++) {  
    scanf("%d", (ptr + i));  
}  
printf("The elements of the array are:\n");  
for (i = 0; i < n; i++) {  
    printf("%d ", *(ptr + i));  
}  
printf("\n");  
return 0;  
}
```

Output:



```
D:\Desktop\C(MCA)\39.exe  ×  +  ▾  
Enter the number of elements in the array: 5  
Enter the elements of the array:  
1  
2  
3  
4  
5  
The elements of the array are:  
1 2 3 4 5  
  
-----  
Process exited after 5.782 seconds with return value 0  
Press any key to continue . . . |
```

// Question : Write a program to copy one array to another using pointer.

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

```
int main() {
```

```
    int n, i;
```

```
    int *sourcePtr, *destPtr;
```

```
    printf("Enter the number of elements in the array: ");
```

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

```
    int source[n], destination[n];
```

```
    sourcePtr = source;
```

```
    destPtr = destination;
```

```
    printf("Enter the elements of the source array:\n");
```

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

```
        scanf("%d", (sourcePtr + i));
```

```
    }
```

```

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

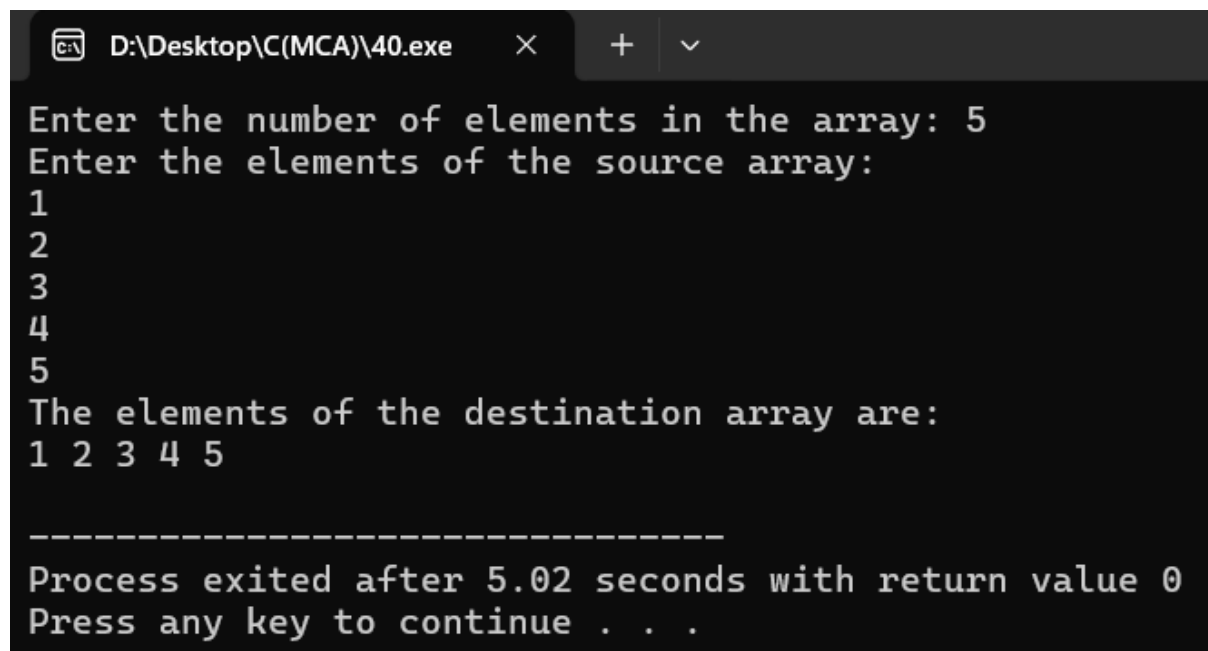
printf("The elements of the destination array are:\n");
for (i = 0; i < n; i++) {
    printf("%d ", *(destPtr + i));
}

printf("\n");

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\40.exe
Enter the number of elements in the array: 5
Enter the elements of the source array:
1
2
3
4
5
The elements of the destination array are:
1 2 3 4 5

-----
Process exited after 5.02 seconds with return value 0
Press any key to continue . . .

```

//Question : Write a program to swap two arrays using pointers.

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

```
void swapArrays(int *arr1, int *arr2, int n) {  
    int temp;  
    for (int i = 0; i < n; i++) {  
        temp = *(arr1 + i); // Store the value of arr1[i] in temp  
        *(arr1 + i) = *(arr2 + i); // Copy the value of arr2[i] to arr1[i]  
        *(arr2 + i) = temp; // Assign the value of temp to arr2[i]  
    }  
}
```

```
int main() {  
    int n;  
  
    // Input the size of the arrays  
    printf("Enter the number of elements in the arrays: ");  
    scanf("%d", &n);  
  
    // Declare two arrays  
    int arr1[n], arr2[n];  
  
    // Input elements for the first array  
    printf("Enter the elements of the first array:\n");  
    for (int i = 0; i < n; i++) {  
        scanf("%d", &arr1[i]);
```

```
}
```

```
// Input elements for the second array
```

```
printf("Enter the elements of the second array:\n");
```

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

```
    scanf("%d", &arr2[i]);
```

```
}
```

```
// Print the arrays before swapping
```

```
printf("\nBefore swapping:\n");
```

```
printf("First array: ");
```

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

```
    printf("%d ", arr1[i]);
```

```
}
```

```
printf("\nSecond array: ");
```

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

```
    printf("%d ", arr2[i]);
```

```
}
```

```
printf("\n");
```

```
// Swap the arrays
```

```
swapArrays(arr1, arr2, n);
```

```
// Print the arrays after swapping
```

```
printf("\nAfter swapping:\n");
```

```
printf("First array: ");
```

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



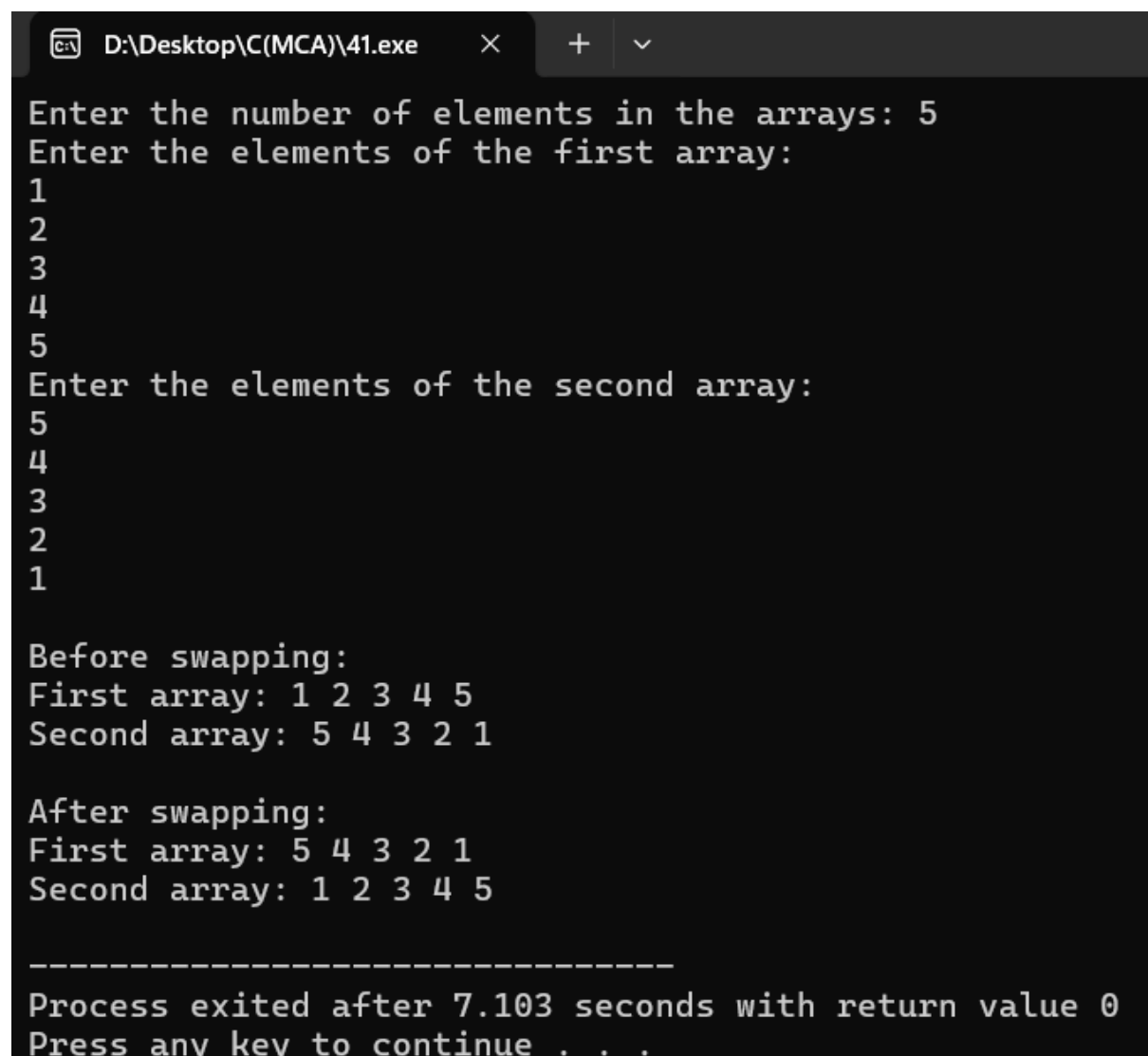
```

        printf("%d ", arr1[i]);
    }
    printf("\nSecond array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr2[i]);
    }
    printf("\n");

    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\41.exe
Enter the number of elements in the arrays: 5
Enter the elements of the first array:
1
2
3
4
5
Enter the elements of the second array:
5
4
3
2
1

Before swapping:
First array: 1 2 3 4 5
Second array: 5 4 3 2 1

After swapping:
First array: 5 4 3 2 1
Second array: 1 2 3 4 5

-----
Process exited after 7.103 seconds with return value 0
Press any key to continue . . .

```

// Question: Write a program to reverse an array using pointers.

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

```
void reverseArray(int *arr, int n) {
```

```
    int *start = arr;
```

```
    int *end = arr + n - 1;
```

```
    int temp;
```

```
    while (start < end) {
```

```
        temp = *start;
```

```
        *start = *end;
```

```
        *end = temp;
```

```
        start++;
```

```
        end--;
```

```
    }
```

```
}
```

```
int main() {
```

```
    int n, i;
```

```
    printf("Enter the number of elements in the array: ");
```

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

```
int arr[n];
```

```
printf("Enter the elements of the array:\n");
```

```
for (i = 0; i < n; i++) {  
    scanf("%d", &arr[i]);  
}
```

```
printf("Original array: ");
```

```
for (i = 0; i < n; i++) {  
    printf("%d ", arr[i]);  
}
```

```
printf("\n");
```

```
reverseArray(arr, n);
```

```
printf("Reversed array: ");
```

```
for (i = 0; i < n; i++) {  
    printf("%d ", arr[i]);  
}
```

```
printf("\n");
```

```
    return 0;
}
```

Output:

```

D:\Desktop\C(MCA)\42.exe
Enter the number of elements in the array: 5
Enter the elements of the array:
1
2
3
4
5
Original array: 1 2 3 4 5
Reversed array: 5 4 3 2 1

-----
Process exited after 4.413 seconds with return value 0
Press any key to continue . . .
```

/*Write a program to search an element in array using pointers.*/

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

```
int main()
```

```
{
```

```
    int arr[] = {10, 20, 30, 40, 50};
```

```
    int size = sizeof(arr) / sizeof(arr[0]); // Calculate the size of the
array
```

```
    int *ptr = arr; // Pointer to the first element of the array
```

```
    int searchElement, found = 0;
```

```
    printf("Array:\n");
```

```
    for (int i = 0; i < size; i++) {
```

```
        printf("%d\t", arr[i]);
```

```
    }
```

```

// Ask the user for the element to search
printf("\nEnter the element to search: ");
scanf("%d", &searchElement);

// Search for the element using pointer
for (int i = 0; i < size; i++) {
    if (*(ptr + i) == searchElement) { // Dereferencing pointer
        printf("Element %d found at index %d\n", searchElement,
i);
        found = 1;
        break;
    }
}
// If the element is not found
if (!found) {
    printf("Element %d not found in the array.\n",
searchElement);
}
return 0;
}

```

Output:

```
D:\Desktop\C(MCA)\43.exe  X  +  v
Array:
10      20      30      40      50
Enter the element to search: 40
Element 40 found at index 3

-----
Process exited after 4.177 seconds with return value 0
Press any key to continue . . . |
```

/*Write a program to add two 2 X 2 matrix using pointers.*/

#include <stdio.h>

int main() {

// Declare two 2x2 matrices and the result matrix

int matrix1[2][2], matrix2[2][2], result[2][2];

int *ptr1, *ptr2, *ptr_result;

// Set pointers to point to the first element of the matrices

ptr1 = &matrix1[0][0];

ptr2 = &matrix2[0][0];

ptr_result = &result[0][0];

// Input elements for the first matrix

printf("Enter elements of the first 2x2 matrix:\n");

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

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

printf("Enter element [%d][%d]: ", i, j);

**scanf("%d", (ptr1 + i * 2 + j)); // Accessing matrix using
pointer arithmetic**

}

```
}
```

```
// Input elements for the second matrix
```

```
printf("Enter elements of the second 2x2 matrix:\n");
```

```
for (int i = 0; i < 2; i++) {
```

```
    for (int j = 0; j < 2; j++) {
```

```
        printf("Enter element [%d][%d]: ", i, j);
```

```
        scanf("%d", (ptr2 + i * 2 + j)); // Accessing matrix using  
pointer arithmetic
```

```
    }
```

```
}
```

```
// Adding the two matrices and storing the result
```

```
for (int i = 0; i < 2; i++) {
```

```
    for (int j = 0; j < 2; j++) {
```

```
        *(ptr_result + i * 2 + j) = *(ptr1 + i * 2 + j) + *(ptr2 + i * 2 + j);
```

```
// Pointer addition
```

```
    }
```

```
}
```

```
// Display the result
```

```
printf("The sum of the two matrices is:\n");
```

```
for (int i = 0; i < 2; i++) {
```

```
    for (int j = 0; j < 2; j++) {
```

```
        printf("%d ", *(ptr_result + i * 2 + j)); // Accessing the result  
matrix using pointer
```

```
    }
```

```
printf("\n");
```

```

    }

    return 0;
}

```

Output:

```

D:\Desktop\C(MCA)\44.exe
Enter elements of the first 2x2 matrix:
Enter element [0][0]: 1
Enter element [0][1]: 2
Enter element [1][0]: 3
Enter element [1][1]: 4
Enter elements of the second 2x2 matrix:
Enter element [0][0]: 4
Enter element [0][1]: 3
Enter element [1][0]: 2
Enter element [1][1]: 1
The sum of the two matrices is:
5 5
5 5

-----
Process exited after 10.23 seconds with return value 0
Press any key to continue . . .

```

/*Write a program to multiply two 2 X 2 matrix using pointers*/

#include <stdio.h>

int main() {

int A[2][2], B[2][2], C[2][2];

int *pA = &A[0][0]; // Pointer to the first element of matrix A

int *pB = &B[0][0]; // Pointer to the first element of matrix B


```
int *pC = &C[0][0]; // Pointer to the first element of matrix C
```

```
// Input elements for matrix A
```

```
printf("Enter the elements of matrix A (2x2):\n");
```

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 2; j++) {  
        scanf("%d", pA++);  
    }  
}
```

```
// Input elements for matrix B
```

```
printf("Enter the elements of matrix B (2x2):\n");
```

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 2; j++) {  
        scanf("%d", pB++);  
    }  
}
```

```
// Matrix multiplication (C = A * B)
```

```
pA = &A[0][0]; // Reset pointer to the start of matrix A
```

```
pB = &B[0][0]; // Reset pointer to the start of matrix B
```

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 2; j++) {  
        *(pC + i * 2 + j) = 0; // Initialize the result matrix  
        for (int k = 0; k < 2; k++) {  
            *(pC + i * 2 + j) += *(pA + i * 2 + k) * *(pB + k * 2 + j);
```

```

    }
}
}

// Display the result matrix
printf("Result matrix C (A * B):\n");
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 2; j++) {
        printf("%d ", *(pC + i * 2 + j));
    }
    printf("\n");
}

return 0;
}

```

Output:

```
D:\Desktop\C(MCA)\45.exe  ×  +  ∨

Enter the elements of matrix A (2x2):
1
2
3
4
Enter the elements of matrix B (2x2):
5
4
3
2
Result matrix C (A * B):
11 8
27 20

-----
Process exited after 10.68 seconds with return value 0
Press any key to continue . . . |
```

/*Write a program to find length of string using pointers*/

#include <stdio.h>

```
int main() {  
    char str[100];  
    char *ptr;  
    int length = 0;  
  
    // Taking input from the user  
    printf("Enter a string: ");  
    gets(str);  
  
    // Pointer initialization  
    ptr = str;
```

```

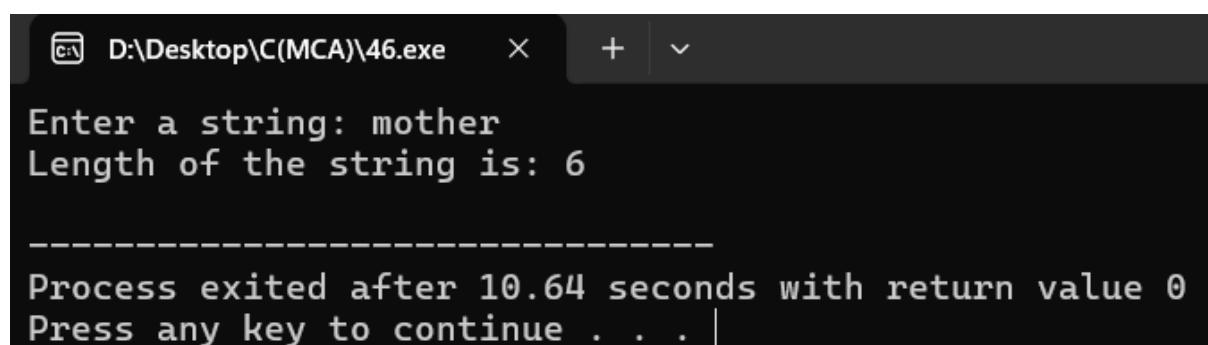
// Traverse the string until the null-terminator is found
while (*ptr != '\0') {
    length++;
    ptr++; // Move to the next character
}

// Output the length of the string
printf("Length of the string is: %d\n", length);

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\46.exe
Enter a string: mother
Length of the string is: 6

-----
Process exited after 10.64 seconds with return value 0
Press any key to continue . . .

```

/*Write a program to copy one string to another using pointer.*/

#include <stdio.h>

```
int main() {
```

```
    char source[] = "Suchorit Saha"; // Source string
```

```
    char destination[50]; // Destination string (large enough to hold
the copied string)
```

```
    // Pointer to source and destination
```

```

char *src = source;

char *dest = destination;

// Copying each character from source to destination using
pointers
while (*src != '\0') {
    *dest = *src; // Copy character from source to destination
    src++;      // Move the source pointer to the next character
    dest++;     // Move the destination pointer to the next position
}

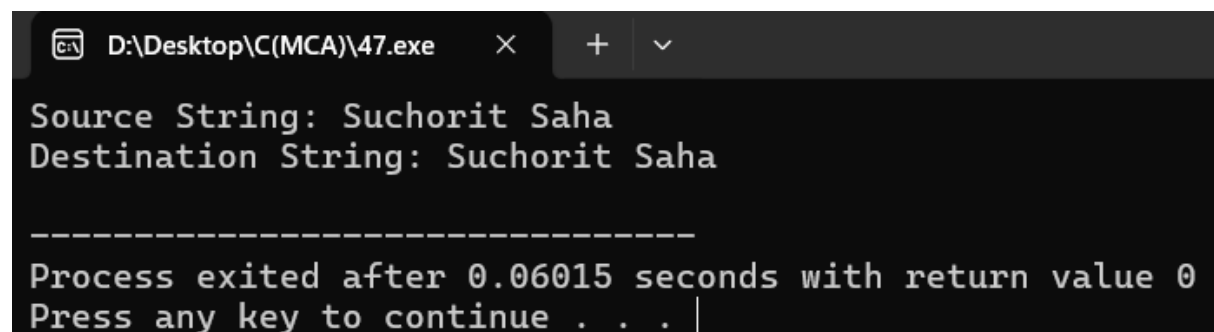
*dest = '\0'; // Null-terminate the destination string

// Print both strings to verify the copy
printf("Source String: %s\n", source);
printf("Destination String: %s\n", destination);

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\47.exe
Source String: Suchorit Saha
Destination String: Suchorit Saha

-----
Process exited after 0.06015 seconds with return value 0
Press any key to continue . . . |

```

/*Write a program to concatenate two strings using pointers*/

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

```
void concatenateStrings(char *str1, char *str2) {
```

```
    // Move the pointer 'str1' to the end of the first string
```

```
    while (*str1 != '\0') {
```

```
        str1++; // Increment the pointer to the next character
```

```
    }
```

```
    // Now, copy the characters of str2 to the end of str1
```

```
    while (*str2 != '\0') {
```

```
        *str1 = *str2; // Copy the character from str2 to str1
```

```
        str1++;      // Move the pointer in str1
```

```
        str2++;      // Move the pointer in str2
```

```
    }
```

```
    // add the null terminator at the end of the concatenated string
```

```
    *str1 = '\0';
```

```
}
```

```
int main() {
```

```
    char str1[100], str2[50];
```

```
    // Input two strings
```

```
    printf("Enter first string: ");
```

```
    gets(str1);
```

```
    printf("Enter second string: ");
```

```

    gets(str2);

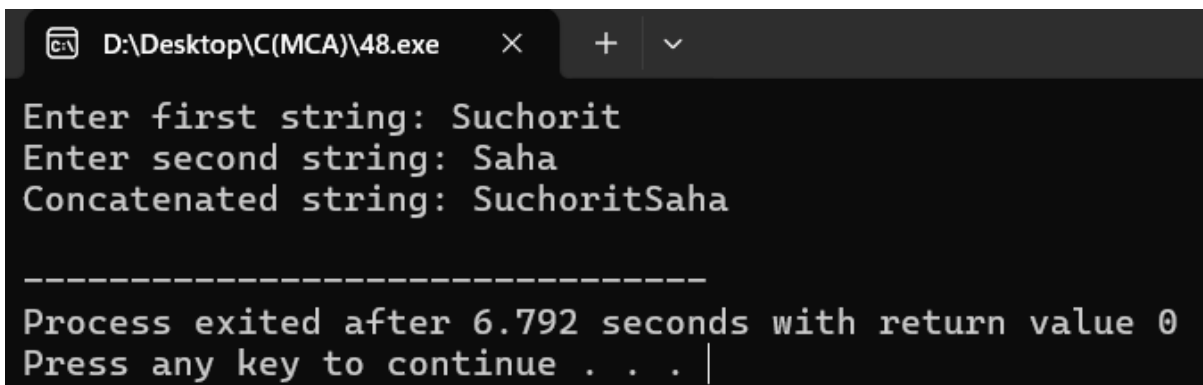
    // Concatenate the strings using pointers
    concatenateStrings(str1, str2);

    // Print the concatenated result
    printf("Concatenated string: %s\n", str1);

    return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\48.exe
Enter first string: Suchorit
Enter second string: Saha
Concatenated string: SuchoritSaha
-----
Process exited after 6.792 seconds with return value 0
Press any key to continue . . . |

```

/*Write a program to compare two strings using pointers*/

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

```

int compareStrings(const char *str1, const char *str2) {
    // Loop through both strings using pointers
    while (*str1 != '\0' && *str2 != '\0') {
        if (*str1 != *str2) {

```

```

        return 0; // Strings are not equal
    }
    str1++;
    str2++;
}

// Check if both strings are of the same length
if (*str1 == '\0' && *str2 == '\0') {
    return 1; // Strings are equal
}

return 0; // Strings are not equal, one string is longer
}

int main() {
    char str1[] = "Hello, World!";
    char str2[] = "Hello, World!";
    char str3[] = "Hello, C!";

    printf("str1->%s str2->%s str3->%s",str1,str2,str3);
    if (compareStrings(str1, str2)) {
        printf("\nstr1 and str2 are equal.\n");
    } else {
        printf("str1 and str2 are not equal.\n");
    }

    if (compareStrings(str1, str3)) {

```



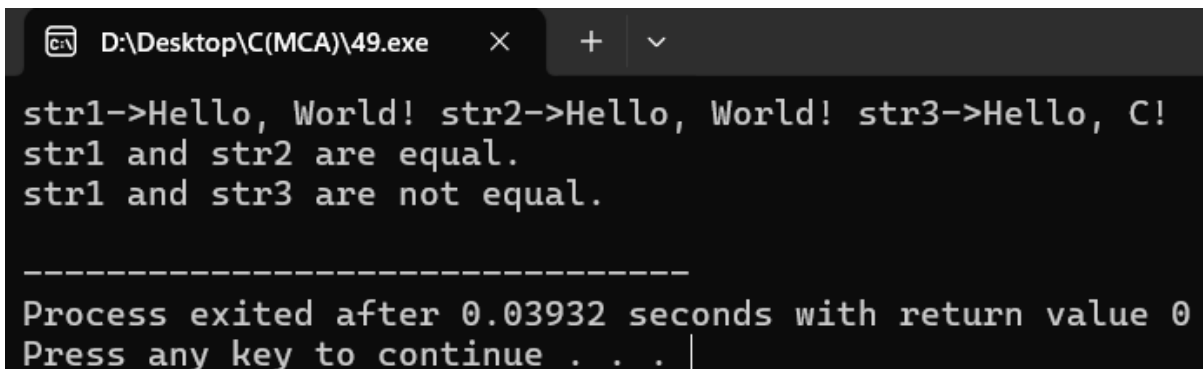
```

    printf("str1 and str3 are equal.\n");
} else {
    printf("str1 and str3 are not equal.\n");
}

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\49.exe
str1->Hello, World! str2->Hello, World! str3->Hello, C!
str1 and str2 are equal.
str1 and str3 are not equal.

-----
Process exited after 0.03932 seconds with return value 0
Press any key to continue . . . |

```

/*C Program to list all files and sub-directories in a directory*/

```

#include <stdio.h>
#include <stdlib.h>
#include <dirent.h> // For directory functions
#include <sys/stat.h> // For file status

void listDirectory(const char *path) {
    struct dirent *entry;
    DIR *dir = opendir(path);

    if (dir == NULL) {
        perror("Unable to read directory");
        return;
    }
}

```

```

}

printf("Contents of directory: %s\n", path);
while ((entry = readdir(dir)) != NULL) {
    // Skip the current and parent directory entries
    if (entry->d_name[0] == '.' &&
        (entry->d_name[1] == '\0' || (entry->d_name[1] == '.' &&
entry->d_name[2] == '\0')))) {
        continue;
    }

    // Print the name of the file or directory
    printf("%s", entry->d_name);

    // Check if it's a directory
    struct stat fileStat;
    char fullPath[1024];
    snprintf(fullPath, sizeof(fullPath), "%s/%s", path, entry-
>d_name);

    if (stat(fullPath, &fileStat) == 0 && S_ISDIR(fileStat.st_mode))
    {
        printf(" [DIR]");
    }
    printf("\n");
}

closedir(dir);

```

```
}
```

```
int main(int argc, char *argv[]) {  
    const char *path;  
  
    if (argc > 1) {  
        path = argv[1]; // Use user-provided path  
    } else {  
        path = "."; // Default to current directory  
    }  
  
    listDirectory(path);  
  
    return 0;  
}
```

Output:

```
D:\Desktop\C(MCA)\50.exe  ×  +  ∨  
Contents of directory: .  
.git [DIR]  
1.cpp  
1.exe  
10.cpp  
10.exe  
11.cpp  
11.exe  
12.cpp  
12.exe  
13.cpp  
13.exe  
14.cpp  
14.exe  
15.cpp  
15.exe  
16.cpp  
16.exe  
17.cpp  
17.exe  
18.cpp  
18.exe  
19.cpp  
19.exe  
2.cpp  
2.exe  
20.cpp  
20.exe  
21.cpp  
21.exe  
22.cpp  
22.exe  
23.cpp  
23.exe  
24.cpp  
24.exe  
25.cpp  
25.exe  
26.cpp  
26.exe  
27.cpp
```

/*C Program to count number of lines in a file*/

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

```
int main() {
```

```
    FILE *file;
```

```
    char ch;
```

```
    int line_count = 1;
```

```
    // Open the file in read mode
```

```
    file = fopen("example.txt", "r");
```

```
    if (file == NULL) {
```

```
        // Check if the file was opened successfully
```

```
        printf("Error opening file.\n");
```

```
        return 1;
```

```
    }
```

```
    // Read the file character by character
```

```
    while ((ch = fgetc(file)) != EOF) {
```

```
        if (ch == '\n') {
```

```
            line_count++; // Increment the line count whenever a  
newline character is encountered
```

```
        }
```

```
    }
```

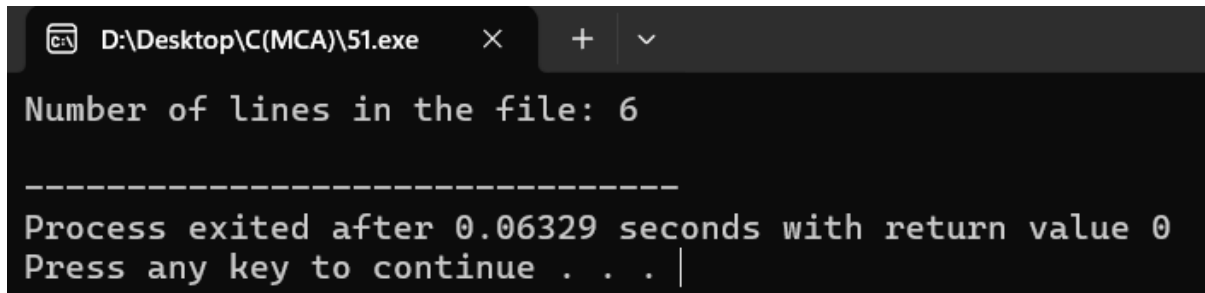
```
    // Close the file
```

```
    fclose(file);
```

```
printf("Number of lines in the file: %d\n", line_count);

return 0;
}
```

Output:



```
D:\Desktop\C(MCA)\51.exe
Number of lines in the file: 6
-----
Process exited after 0.06329 seconds with return value 0
Press any key to continue . . . |
```

/*C Program to print contents of file*/

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

```
int main() {
```

```
    FILE *file;
```

```
    char ch;
```

```
    // Open the file in read mode
```

```
    file = fopen("example.txt", "r");
```

```
    // Check if file opened successfully
```

```
    if (file == NULL) {
```

```
        printf("Could not open the file.\n");
```

```
        return 1;
```

```
    }
```

```
    // Read and print the content of the file character by character
```

```

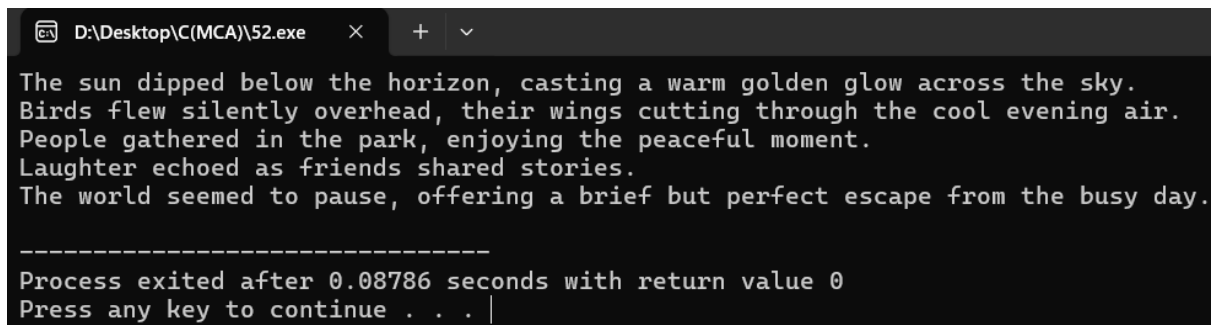
while ((ch = fgetc(file)) != EOF) {
    putchar(ch);
}

// Close the file after reading
fclose(file);

return 0;
}

```

Output:



```

D:\Desktop\C(MCA)\52.exe
The sun dipped below the horizon, casting a warm golden glow across the sky.
Birds flew silently overhead, their wings cutting through the cool evening air.
People gathered in the park, enjoying the peaceful moment.
Laughter echoed as friends shared stories.
The world seemed to pause, offering a brief but perfect escape from the busy day.

-----
Process exited after 0.08786 seconds with return value 0
Press any key to continue . . . |

```

/*C Program to copy contents of one file to another file*/

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

```

int main() {
    FILE *sourceFile, *destinationFile;
    char ch;

    // Open the source file in read mode
    sourceFile = fopen("example.txt", "r");
    if (sourceFile == NULL) {
        printf("Error opening source file.\n");
    }
}

```

```

    return 1;
}

// Open the destination file in write mode
destinationFile = fopen("destination.txt", "w");
if (destinationFile == NULL) {
    printf("Error opening destination file.\n");
    fclose(sourceFile); // Don't forget to close the source file
    before returning
    return 1;
}

// Read each character from the source file and write it to the
destination file
while ((ch = fgetc(sourceFile)) != EOF) {
    fputc(ch, destinationFile);
}

printf("File copied successfully.\n");



// Close both files
fclose(sourceFile);
fclose(destinationFile);

return 0;
}

```

Output:


```
D:\Desktop\C(MCA)\53.exe  X  +  v
File copied successfully.
-----
Process exited after 0.07182 seconds with return value 0
Press any key to continue . . . |
```

 destination	25-11-2024 23:20	Text Document	1 KB
 example	25-11-2024 23:17	Text Document	1 KB

/*C Program to merge contents of two files into a third file*/

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

```
int main() {
```

```
    FILE *file1, *file2, *file3;
```

```
    char ch;
```

```
    // Open the first file in read mode
```

```
    file1 = fopen("example.txt", "r");
```

```
    if (file1 == NULL) {
```

```
        printf("Could not open file1.txt\n");
```

```
        return 1;
```

```
    }
```

```
    // Open the second file in read mode
```

```
    file2 = fopen("destination.txt", "r");
```

```
    if (file2 == NULL) {
```

```
        printf("Could not open file2.txt\n");
```

```
        fclose(file1);
```

```

    return 1;
}

// Open the third file in write mode (creates or overwrites)
file3 = fopen("merged_file.txt", "w");
if (file3 == NULL) {
    printf("Could not open merged_file.txt\n");
    fclose(file1);
    fclose(file2);
    return 1;
}

// Copy contents of the first file to the third file
while ((ch = fgetc(file1)) != EOF) {
    fputc(ch, file3);
}

// Copy contents of the second file to the third file
while ((ch = fgetc(file2)) != EOF) {
    fputc(ch, file3);
}

printf("Files    have    been    merged    successfully    into
merged_file.txt\n");

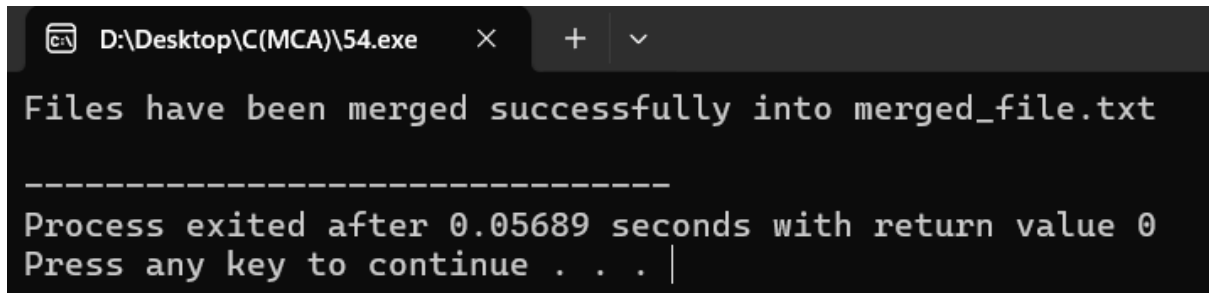
// Close all the files
fclose(file1);
fclose(file2);

```

```
fclose(file3);

return 0;
}
```

Output:



```
D:\Desktop\C(MCA)\54.exe
Files have been merged successfully into merged_file.txt
-----
Process exited after 0.05689 seconds with return value 0
Press any key to continue . . . |
```

/* C program to delete a file */

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

```
int main() {
```

```
    const char *filename = "destination.txt"; // Specify the file you
    want to delete
```

```
    // Attempt to delete the file
```

```
    if (remove(filename) == 0) {
```

```
        printf("File %s deleted successfully.\n", filename);
```

```
    } else {
```

```
        printf("Error occurred in deleting the file %s.\n", filename);
```

```
    }
```

```
    return 0;
```

```
}
```

Output:



D:\Desktop\C(MCA)\55.exe



File destination.txt deleted successfully.

Process exited after 0.06096 seconds with return value 0

Press any key to continue . . . |