**1.FACTORIAL WITHOUT USING RECURSION**

#include<stdio.h>

int fact (int num){

if(num==0){

return 1;

}

else{

return num\*fact(num-1);

}

}

int main(){

int num ;

printf("enter the number:");

scanf("%d",&num);

printf("factorial of %d is %d",num,fact(num));

}

INPUT: enter the number:4

OUTPUT: factorial of 4 is 24

**2.FACTORIAL USING RECURSION**

#include<stdio.h>

int fact (int num){

if(num==0){

return 1;

}

else{

return num\*fact(num-1);

}

}

int main(){

int num;

printf("Enter the number");

scanf("%d",&num);

printf("factorial of %d is %d", num, fact(num));

return 0;

}

INPUT: Enter the number 7

OUTPUT: factorial of 7 is 5040

**3. FIBONACCI WITHOUT USING RECURSION**

#include <stdio.h>

int main() {

int i, n;

int t1 = 0, t2 = 1;

int next = t1 + t2;

printf("Enter the number of terms: ");

scanf("%d", &n);

printf("Fibonacci Series: %d, %d, ", t1, t2);

for (i = 3; i <= n; ++i) {

t1 = t2;

t2 = next;

next= t1 + t2;

printf("%d, ", next);

}

return 0;

}

INPUT: Enter the number of terms: 7

OUTPUT: Fibonacci Series: 0, 1, 2, 3, 5, 8, 13,

**4.FIBONACCI USING RECURSION**

#include<stdio.h>

void fib(int n){

int t1=0,t2=1,next;

for(int i=3;i<=n;i++){

next=t1+t2;

t1=t2;

t2=next;

printf("%d , ",next);

}

}

int main(){

int n;

printf("enter the number: ");

scanf("%d",&n);

printf("fibonacci series : %d , %d ,",0,1);

fib(n);

}

INPUT: enter the number: 4

OUTPUT: fibonacci series : 0 , 1 ,1 , 2

**5. FIND EVEN AND ODD NUMBERS IN AN ARRAY**

#include <stdio.h>

int main() {

int arr[10]={1,2,3,4,5,6,7,8,9,10};

int i;

printf("\nEven numbers are:\n");

for(i=0;i<10;i++){

if(arr[i] % 2 == 0)

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

}

printf("\nOdd numbers are:\n");

for(i=0;i<10;i++){

if(arr[i] % 2 != 0)

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

}

return 0;

}

**OUTPUT:**

Even numbers are:

2 4 6 8 10

Odd numbers are:

1 3 5 7 9

**6. MATRIX MULTIPLICATION**

#include<stdio.h>

int main() {

int a[10][10], b[10][10], c[10][10], n, i, j, k;

printf("Enter the value of N : ");

scanf("%d", & n);

printf("Enter the elements of Matrix-A: \n");

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

for (j = 0; j < n; j++) {

scanf("%d", & a[i][j]);

}

}

printf("Enter the elements of Matrix-B: \n");

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

for (j = 0; j < n; j++) {

scanf("%d", & b[i][j]);

}

}

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

for (j = 0; j < n; j++) {

c[i][j] = 0;

for (k = 0; k < n; k++) {

c[i][j] =c[i][j] + a[i][k] \* b[k][j];

}

}

}

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

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

for (j = 0; j < n; j++) {

printf("%d\t", c[i][j]);

}

printf("\n");

}

return 0;

}

INPUT: Enter the value of N: 2

Enter the elements of Matrix-A:

2

4

6

8

Enter the elements of Matrix-B:

1

3

5

7

OUTPUT: The product of the two matrices is:

22 34

46 74

**7.PROGRAM FOR STRING PALINDROME**

#include<stdio.h>

#include<string.h>

int main(){

int i,temp=1;

char s[100];

scanf("%s",&s);

int n= sizeof (s)/ sizeof(s[0]);

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

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

temp=0;

break;

}

}

if(i!=1){

printf("%s is a palindrome:",s);

}

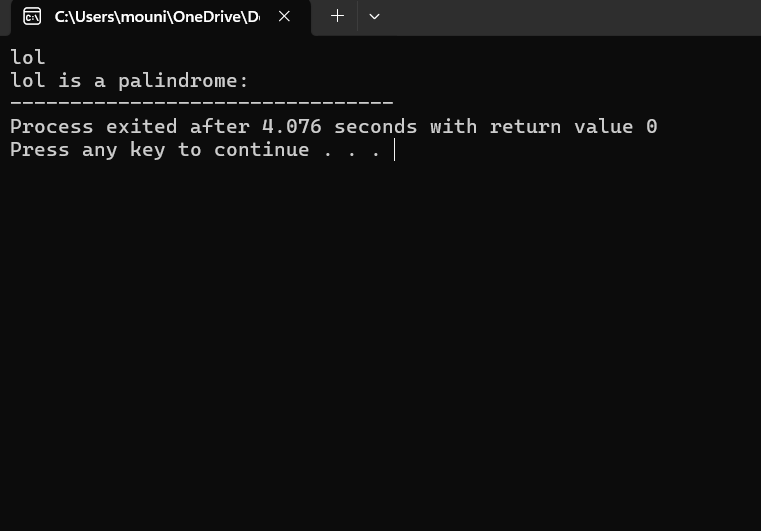
else{

printf("%s not a palindrome:",s);

}

}

INPUT: mom

OUTPUT: mom is a palindrome

**8.PROGRAM FOR NUMBER PALINDROME**

int main(){

int n,m,sum=0;

scanf("%d",&n);

int temp= n;

while(n>0)

{

m= n%10;

sum = (sum\*10)+m;

n=n/10;

}

if(sum == temp)

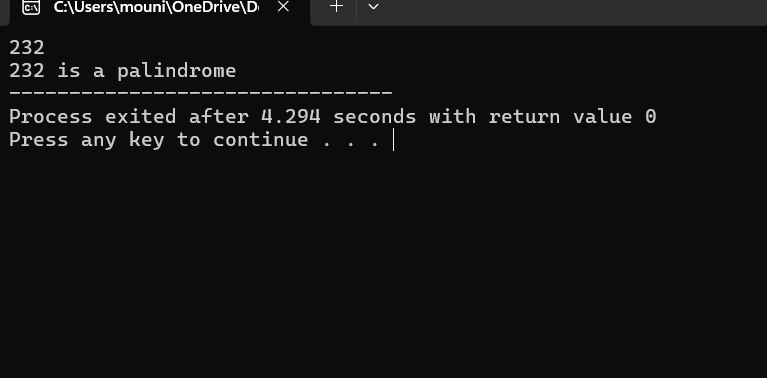
printf("%d is a palindrome",temp);

else

printf("%d is not a palindrome",temp);

return 0;

}



**9.PROGRAM FOR LINEAR SEARCH**

#include<stdio.h>

int main(){

int a[]={12,24,36,52,108,136};

int target=52, i, count=0, temp;

int n=sizeof(a)/sizeof(a[0]);

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

if(a[i]==target){

count=1;

temp=i;

}

}

if(count==1){

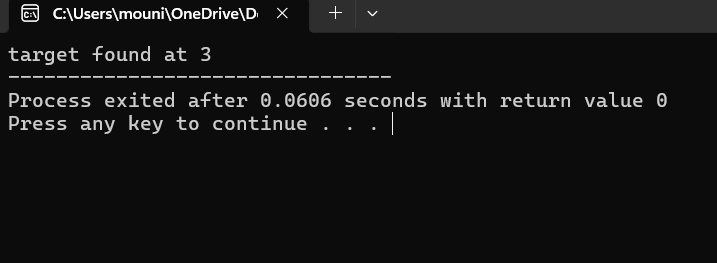
printf("target found at %d",temp);

}

else

printf("target not found ");

}



**10. PROGRAM FOR BINARY SEARCH**

#include <stdio.h>

int binarySearch(int a[], int low, int high, int target)

{

int mid;

if(high >= low)

{

mid = (low+ high)/2;

if(a[mid] == target)

{

return mid;

}

else if(a[mid] < target)

{

return binarySearch(a, mid+1,high, target);

}

else

{

return binarySearch(a, low, mid-1, target);

}

}

return -1;

}

int main()

{

int a[] = {11, 14, 25, 30, 40, 41, 52, 57, 70};

int target= 70;

int n = sizeof(a) / sizeof(a[0]);

int res = binarySearch(a, 0,n-1, target);

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

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

printf("%d ", a[i]);

printf("\nElement to be searched is %d", target);

if (res == -1)

printf("\nElement is not present in the array");

else

printf("\nElement is present at %d position of array", res);

}

