**CSA0358-DATA STRUCTURES WITH GRAPH ALGORITHMS**

**DAY-1(08/08/2023)**

**QUESTION 1:**

Write a C program to check whether the given number is even or odd.

**CODE:**

#include<stdio.h>

int main(){

int n;

printf("Enter the number: ");

scanf("%d",&n);

if(n%2==0){

printf("%d is an even number",n);

}

else{

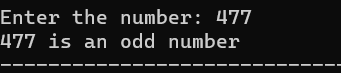
printf("%d is an odd number",n);

}

return 0;

}

**OUTPUT:**



**QUESTION 2:**

Write a C program to find the sum of first n numbers using a for loop.

**CODE:**

#include<stdio.h>

int main(){

int n,sum=0,i;

printf("Enter the number: ");

scanf("%d",&n);

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

sum=sum+i;

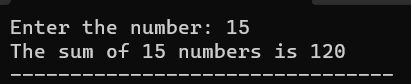
}

printf("The sum of %d numbers is %d",n,sum);

return 0;

}

**OUTPUT:**



**QUESTION 3:**

Write a C program to find the sum of even numbers using a while loop.

**CODE:**

#include<stdio.h>

int main(){

int i,n,even\_sum=0;

printf("Enter the nth number: ");

scanf("%d",&n);

while(i<=n){

if(i%2==0){

even\_sum+=i;

}

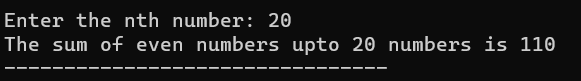
i++;

}

printf("The sum of even numbers upto %d numbers is %d",n,even\_sum);

return 0;

}

**OUTPUT:**

**QUESTION 4:**

Write a C program to reverse a given number.

**CODE:**

#include<stdio.h>

int main(){

int n,rem,rev;

printf("Enter the number: ");

scanf("%d",&n);

while(n!=0){

rem=n%10;

rev=(rev\*10)+rem;

n=n/10;

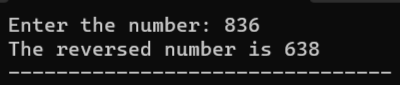
}

printf("The reversed number is %d",rev);

return 0;

}

**OUTPUT:**

****

**QUESTION 5:**

Write a C program to check whether the given number is a palindrome or not.

**CODE:**

#include<stdio.h>

int main(){

int n,rem,rev,temp;

printf("Enter the number: ");

scanf("%d",&n);

temp=n;

while(n!=0){

rem=n%10;

rev=(rev\*10)+rem;

n=n/10;

}

if(rev==temp){

printf("The number is a palindrome");

}

else{

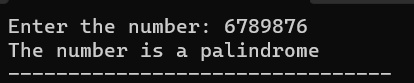
printf("The number is not a palindrome");

}

return 0;

}

**OUTPUT:**

****

**QUESTION 6:**

Write a C program to check whether the given is an armstrong number or not.

**CODE:**

#include<stdio.h>

int main(){

int n,rem,res=0,temp;

printf("Enter a number: ");

scanf("%d",&n);

temp=n;

while(n!=0){

rem=n%10;

res+=rem\*rem\*rem;

n=n/10;

}

if(res==temp){

printf("The number is an armstrong number");

}

else{

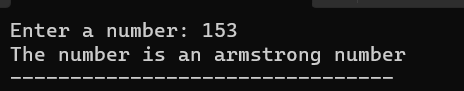
printf("The number is not an armstrong number");

}

return 0;

}

**OUTPUT:**

****

**QUESTION 7:**

Write a C program to find the factorial of the given number without recursion.

**CODE:**

#include<stdio.h>

int main(){

int i,n,fact=1;

printf("Enter the number: ");

scanf("%d",&n);

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

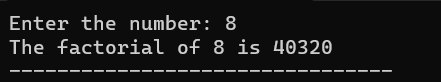
fact=fact\*i;

}

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

}

**OUTPUT:**

****

**QUESTION 8:**

Write a C program to find the factorial of the given number with recursion.

**CODE:**

#include<stdio.h>

int factorial(int n){

if(n==1||n==0){

return 1;

}

else{

return n\*factorial(n-1);

}

}

int main(){

int n,res;

printf("Enter the number: ");

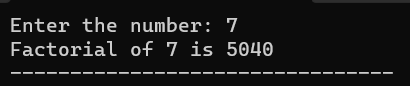
scanf("%d",&n);

res=factorial(n);

printf("Factorial of %d is %d",n,res);

}

**OUTPUT:**

****

**QUESTION 9:**

Write a C program to generate the fibonacci series without using recursion.

**CODE:**

#include<stdio.h>

int main(){

int n,n1=0,n2=1,n3,i;

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

scanf("%d",&n);

printf("The fibonacci series is:\n%d %d ",n1,n2);

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

n3=n1+n2;

n1=n2;

n2=n3;

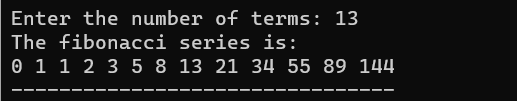
printf("%d ",n3);

}

return 0;

}

**OUTPUT:**

****

**QUESTION 10:**

Write a C program to generate the fibonacci series using recursion.

**CODE:**

#include<stdio.h>

int fibonacci(int n){

if(n<=1){

return n;

}

else{

return fibonacci(n-1)+fibonacci(n-2);

}

}

int main(){

int n,i;

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

scanf("%d",&n);

printf("Fibonacci series upto %d terms is:\n",n);

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

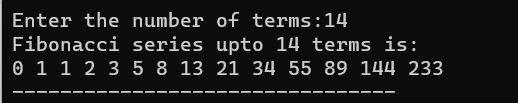
printf("%d ",fibonacci(i));

}

return 0;

}

**OUTPUT:**



**QUESTION 11:**

Write a C program to search a particular element in an array using linear search.

**CODE:**

#include <stdio.h>

int linearSearch(int a[], int n, int val) {

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

{

if (a[i] == val)

return i+1;

}

return -1;

}

int main() {

int n;

int a[100];

printf("Enter the size of the array: ");

scanf("%d",&n);

printf("Enter the elements of the array: ");

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

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

}

int val;

printf("Enter the value to be searched: ");

scanf("%d",&val);

int res = linearSearch(a, n, val);

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

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

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

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

if (res == -1)

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

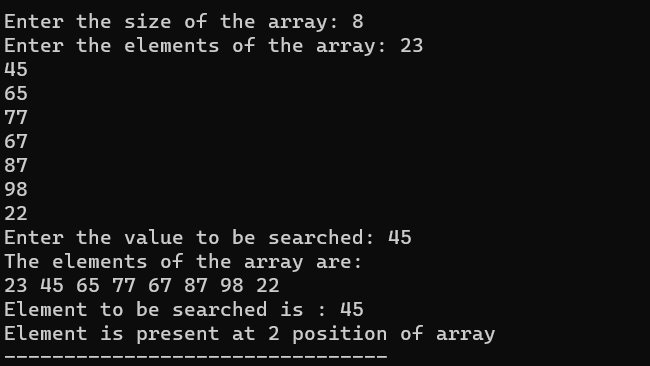
else

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

return 0;

}

**OUTPUT:**

****

**QUESTION 12:**

Write a C program to search a particular element in an array using binary search.

**CODE:**

#include <stdio.h>

int iterativeBinarySearch(int array[], int start\_index, int end\_index, int element){

while (start\_index <= end\_index){

int middle = start\_index + (end\_index- start\_index )/2;

if (array[middle] == element)

return middle;

if (array[middle] < element)

start\_index = middle + 1;

else

end\_index = middle - 1;

}

return -1;

}

int main(void){

int array[] = {1, 4, 7, 9, 16, 56, 70};

int n = 7;

int element = 16;

int found\_index = iterativeBinarySearch(array, 0, n-1, element);

if(found\_index == -1 ) {

printf("Element not found in the array ");

}

else {

printf("Element found at index : %d",found\_index);

}

return 0;

}

**OUTPUT:**

****