DAY-1

DATA STRUCTURES

1.Write a C program to find the given number is odd or even.

#include<stdio.h>

int main()

{

int n;

printf("enter the number");

scanf("%d",&n);

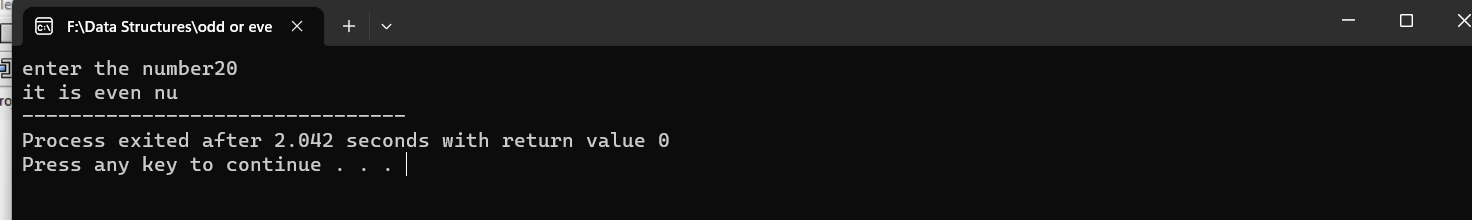
if(n%2==0)

printf("it is even no");

else

printf("it is odd no");

}



2.Write a C program to find sum of natural number using foor loop.

#include<stdio.h>

int main()

{

int i,n,sum;

printf("enert the number");

scanf("%d",&n);

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

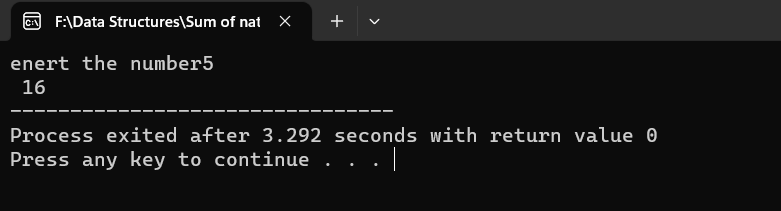
{

sum=sum+i;

}

printf(" %d",sum);

}



3.Write a C program to find sum of even natural number using while loop.

#include<stdio.h>

int main()

{

int i=2,n,sum=0;

printf("enter the number");

scanf("%d",&n);

while(i<=n)

{

if(i%2==0)

{

sum+=i;

}

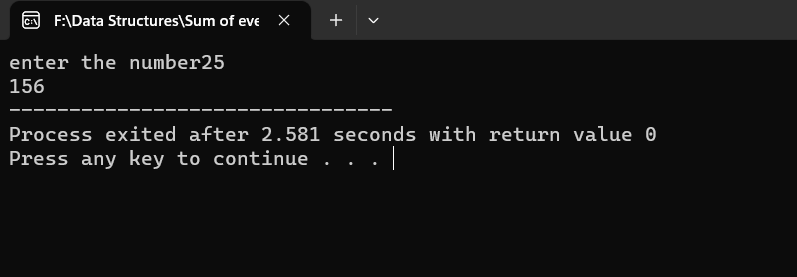
++i;

}

printf("%d",sum);

return 0;

}



4.Write a C program to reverse the given number.

#include<stdio.h>

int main()

{

int n,rev=0,rem;

printf("enter the number");

scanf("%d",&n);

while(n!=0)

{

rem=n%10;

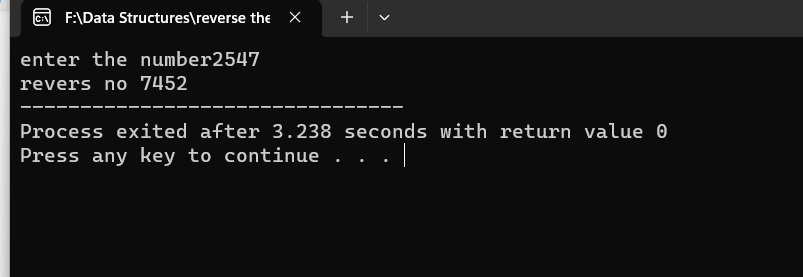
rev=rev\*10+rem;

n=n/10;

}

printf("revers no %d",rev);

}



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

#include<stdio.h>

int main()

{

int orginaln,n,rem,sum=0;

printf("enter the number");

scanf("%d",&n);

orginaln=n;

while(n!=0)

{

rem=n%10;

sum=sum\*10+rem;

n=n/10;

}

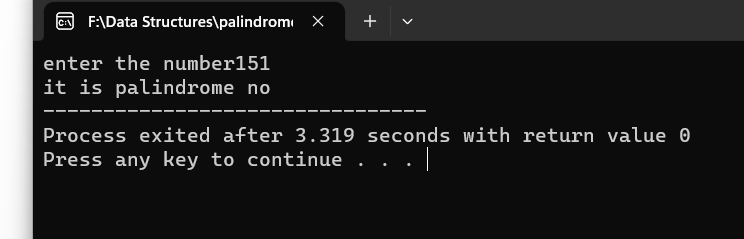
if(orginaln==sum)

printf("it is palindrome no");

else

printf("it is not an palindrome no ");

}



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

#include<stdio.h>

int main()

{

int temp,n,rem,sum=0;

printf("enter the number");

scanf("%d",&n);

temp=n;

while(n>0)

{

rem=n%10;

sum=sum+(rem\*rem\*rem);

n=n/10;

}

if(temp==sum)

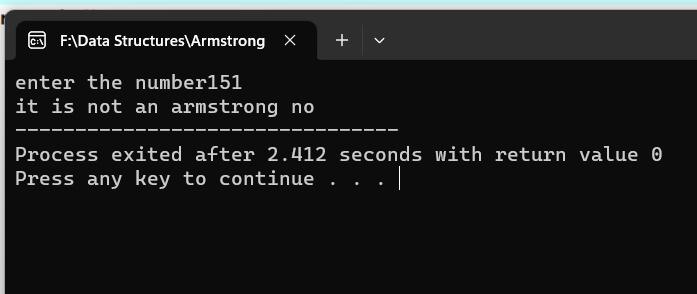
printf("it is armstrong no ");

else

printf("it is not an armstrong no");

return 0;

}



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

#include<stdio.h>

#include<conio.h>

int fact(int number)

{

if(number==0 || number==1)

return 1;

else

return(number \* fact(number-1));

}

int main()

{

int n,f;

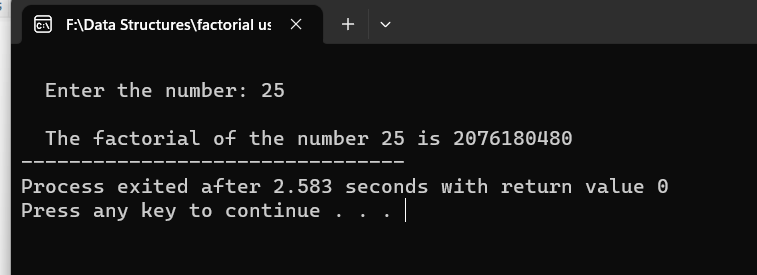
printf("\n Enter the number: ");

scanf("%d",&n);

f=fact(n);

printf("\n The factorial of the number %d is %d",n,f);

}



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

#include<stdio.h>

int main() {

int n, i;

long factorial = 1;

printf("Enter an no ");

scanf("%d",&n);

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

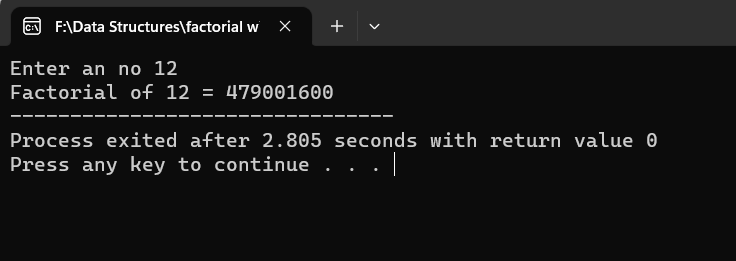
factorial \*= i;

}

printf("Factorial of %d = %llu", n, factorial);

return 0;

}



9.Write a C program to generate Fibonacci series with recursion.

#include <stdio.h>

int Fibonacci(int n)

{

static int n1= 0,n2=1,n3;

if(n > 0){

n3 = n1 + n2;

n1 = n2;

n2 = n3;

printf("%d",n3);

Fibonacci(n-1);

}

}

int main(){

int n;

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

scanf("%d", &n);

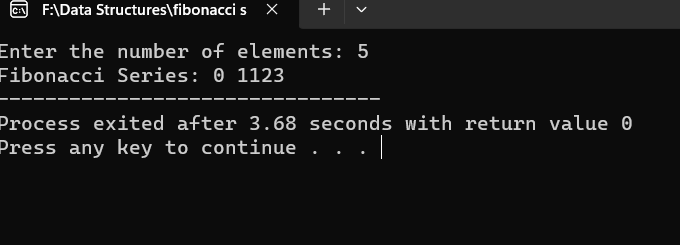
printf("Fibonacci Series: ");

printf("%d %d",0,1);

Fibonacci(n-2);

return 0;

}



10.Write a C program to generate Fibonacci series without recursion.

#include<stdio.h>

int main()

{

int a=0,b=1,result=0,n,i;

printf("enter the number");

scanf("%d",&n);

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

{

printf("%d",a);

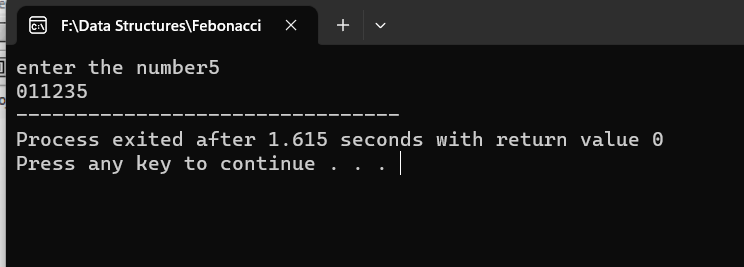
result=a+b;

a=b;

b=result;

}

}



11.Write a C program to search the particular element array using Linear search.

#include <stdio.h>

int linearSearch(int array[], int size, int element) {

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

if(array[i] == element)

return i;

}

return -1;

}

int main() {

int array[] = {1, 3, 5, 7, 9}, element;

int size = sizeof(array) / sizeof(array[0]);

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

scanf("%d", &element);

int searchResult = linearSearch(array, size, element);

if(searchResult != -1)

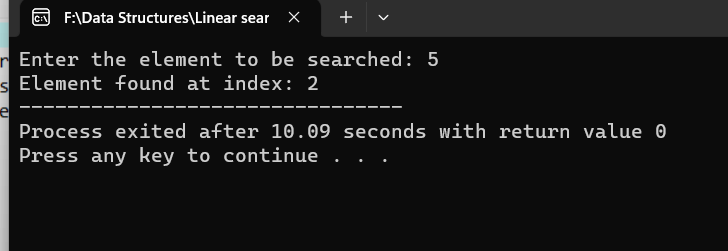
printf("Element found at index: %d", searchResult);

else

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

return 0;

}



12. Write a C program to search the particular element array using Binary search.

#include <stdio.h>

int start = 0, end = size-1;

while(start <= end) {

int mid = (start + end) / 2;

if (array[mid] == element)

return mid;

if (array[mid] < element)

start = mid + 1;

else

end = mid - 1;

}

return -1;

}

int main() {

int array[] = {387,333,331,987,689}, element;

int size = sizeof(array) / sizeof(array[0]);

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

scanf("%d", &element);

int searchResult = binarySearch(array, size, element);

if(searchResult != -1)

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

else

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

return 0;

}

