DATE: 26/03/2022

EXPERIMENT NO. 6

Q. To study and demonstrate Functions and Recursion.

PROGRAM STATEMENTS:

a. Write a C program to create user defined function to reverse string and concatenate.

b. Write a C program to create user defined function to compute surface area and volume of aright circular cylinder. Do not print the output in function body. Do not use global variable. (Use Call by Reference)

c. Write a C program using Recursive Function to compute

1. GCD of two numbers.

2. Find input number is prime or not.

3. Nth Fibonacci terms (number) (n<=40)

THEORY:

1. Functions: The functions contains the set of programming statements enclosed by {}. A function can be called multiple times to provide reusability and modularity to the C program. In other words, we can say that the collection of functions creates a program.

2. Recursion: In the world of programming, when your program lets you call that specific function from inside that function, then this concept of calling the function from itself can be termed as recursion, and the function which makes this possible is called the recursive function.

Program a)

1. Reverse of string:

#include<stdio.h>

void strrev(char str[20]);

int main()

{

char str[20];

printf("Enter the string you want to reverse: ");

scanf("%s",str);

strrev(str);

printf("Reversed string: %s",str);

return 0;

}

void strrev(char str[20])

{

int i=0,len=0;

char temp;

for(i=0;str[i]!='\0';i++)

{

len++;

}

for(i=0;i<len/2;i++)

{

temp=str[i];

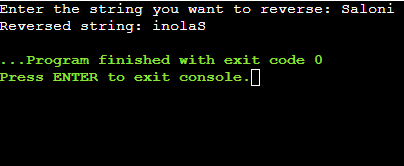
str[i]=str[len-1-i];

str[len-1-i]=temp;

}

}

OUTPUT:



2. Concatenation of two strings:

#include<stdio.h>

void con(char str1[40],char str2[40]);

int main()

{

char str1[40],str2[40];

printf("Enter first string: ");

scanf("%s",str1);

printf("Enter second string: ");

scanf("%s",str2);

con(str1,str2);

printf("New string: %s",str1);

return 0;

}

void con(char str1[40],char str2[40])

{

int i=0,len=0;

for(i=0;str1[i]!='\0';i++)

{

len++;

}

str1[len]=' ';

for(i=0;str2[i]!='\0';i++)

{

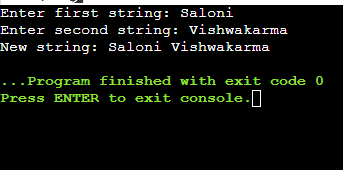
str1[len+i+1]=str2[i];

}

str1[len+i+1]='\0';

}

OUTPUT:



Program b)

#include <stdio.h>

void math(int,int,float\*,float\*);

int main()

{

int r,h;

float area,vol;

printf("Enter the radius and height of the cylinder: ");

scanf("%d %d",&r,&h);

math(r,h,&area,&vol);

printf("Surface area=%f\n",area);

printf("Volume=%f\n",vol);

return 0;

}

void math(int a,int b,float\*c,float\*d)

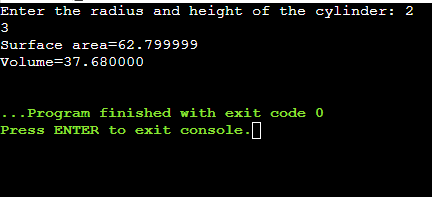
{

\*c=(2\*3.14\*a\*b)+(2\*3.14\*a\*a);

\*d=(3.14\*a\*a\*b);

}

OUTPUT:



Program c)

1. GCD of two numbers

#include<stdio.h>

int gcd(int m,int n);

int main()

{

int m,n,c;

printf("Enter two positive integers: ");

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

c=gcd(m,n);

printf("GCD of %d and %d is %d",m,n,c);

return 0;

}

int gcd(int m,int n)

{

if(n!=0)

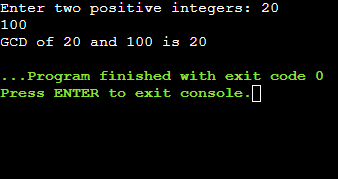
return gcd(n,m%n);

else

return m;

}

OUTPUT:



2. Find input number is prime or not.

#include<stdio.h>

int prime(int,int);

int main()

{

int num,check;

printf("Enter a number: ");

scanf("%d",&num);

check=prime(num,num/2);

if(check==1)

{

printf("%d is a prime number\n",num);

}

else

{

printf("%d is not a prime number\n",num);

}

return 0;

}

int prime(int num,int i)

{

if(i==1)

{

return 1;

}

else

{

if(num%i==0)

{

return 0;

}

else

{

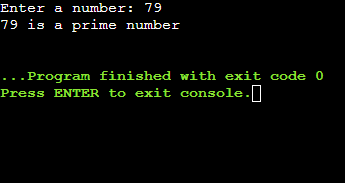
return prime(num,i-1);

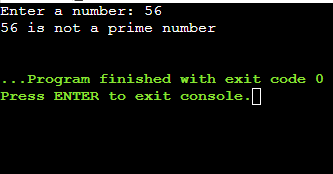
}

}

}

OUTPUT:





3. Nth Fibonacci terms(number) (n<=40)

#include<stdio.h>

int main()

{

int fibo,num;

printf("Enter the Nth term of fibonacci series: ");

scanf("%d",&num);

fibo=fibonacci(num);

printf("\nThe %dth term of fibonacci series is: %d",num,fibo);

}

int fibonacci(int n)

{

if(n==1)

return 1;

else if(n==0)

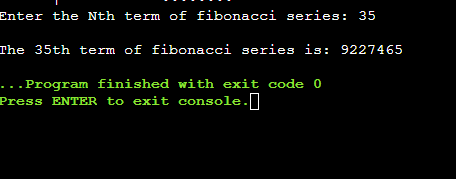
return 0;

else

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

}

OUTPUT:



CONCLUSION: In this practical, we learned the implementation of user defined functions and recursions. We also learned the difference between call by value and call by reference. All the tasks that can be done by loops (iterations) can also be done by recursions and sometimes also vice versa is true.