DATA STRUCTURES LABORATORY

Roll No. :- [20124107]

Q.1. Write a C function to input a number and generate Fibonacci Series. (Function with no arguments and no return value).

#include<stdio.h>

void fibbo(int n){

int a1=0,a2=1,ans=0;

printf("%d %d ",a1,a2);

while(ans<n){

ans=a1+a2;

a1=a2;

a2=ans;

printf("%d ",ans);

}

}

void main(){

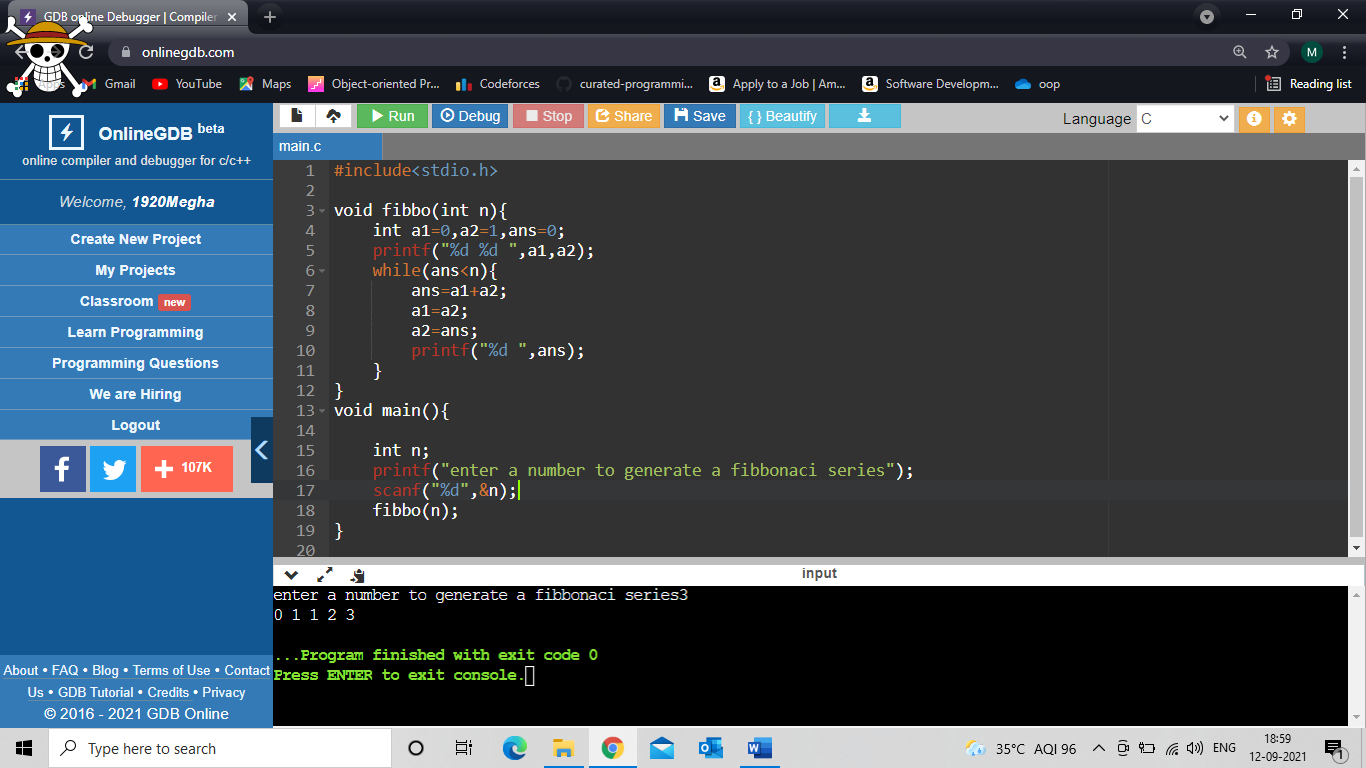
int n;

printf("enter a number to generate a fibbonaci series");

scanf("%d",&n);

fibbo(n);

}



Q.2. Write a C function that returns a random prime number on each function

call. (Function with no arguments and a return value).

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

int rprime(){

srand(time(NULL));

int x= rand();

return x;

}

int main() {

int n,m,flag=0;

printf("this s a random prime number");

again:

n=rprime();

m=n/2;

for(int i=2;i<=m;i++)

{

if(n%i==0)

{

goto again;

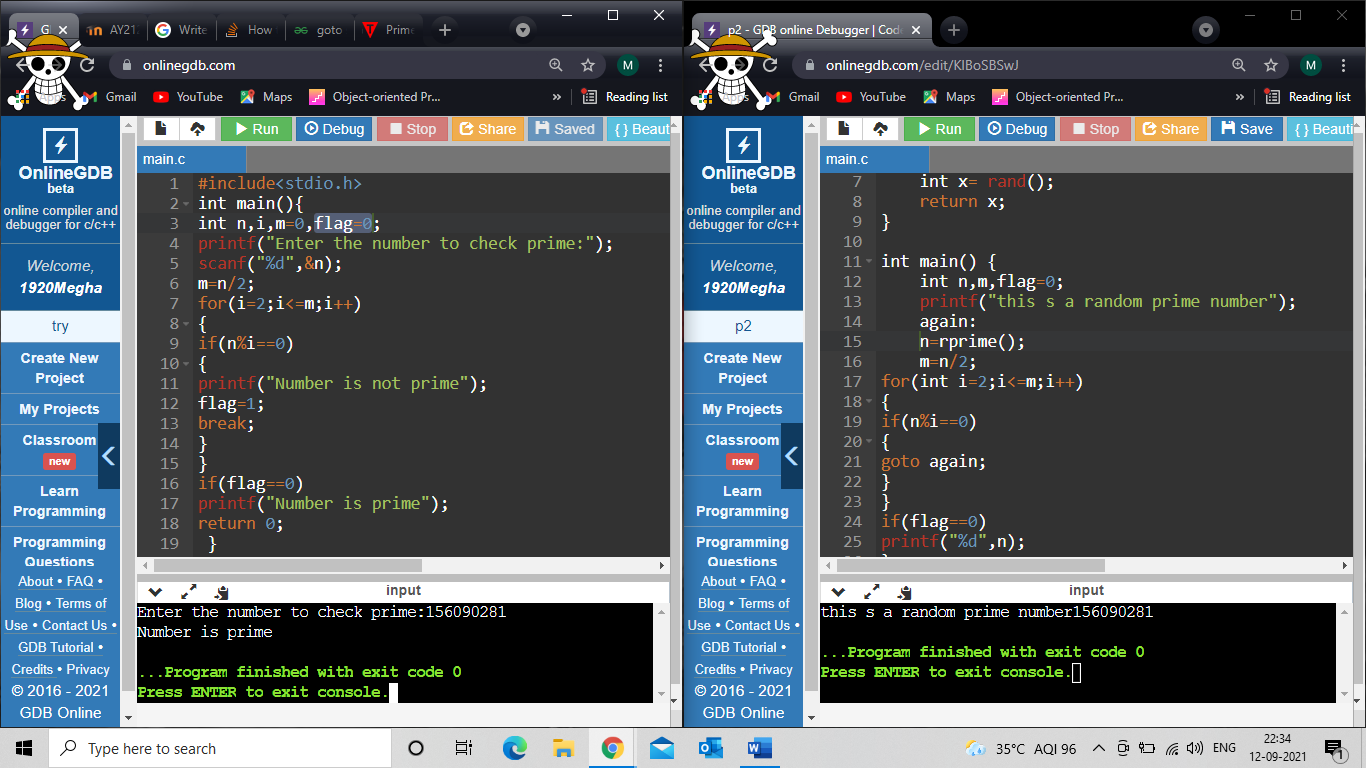
}

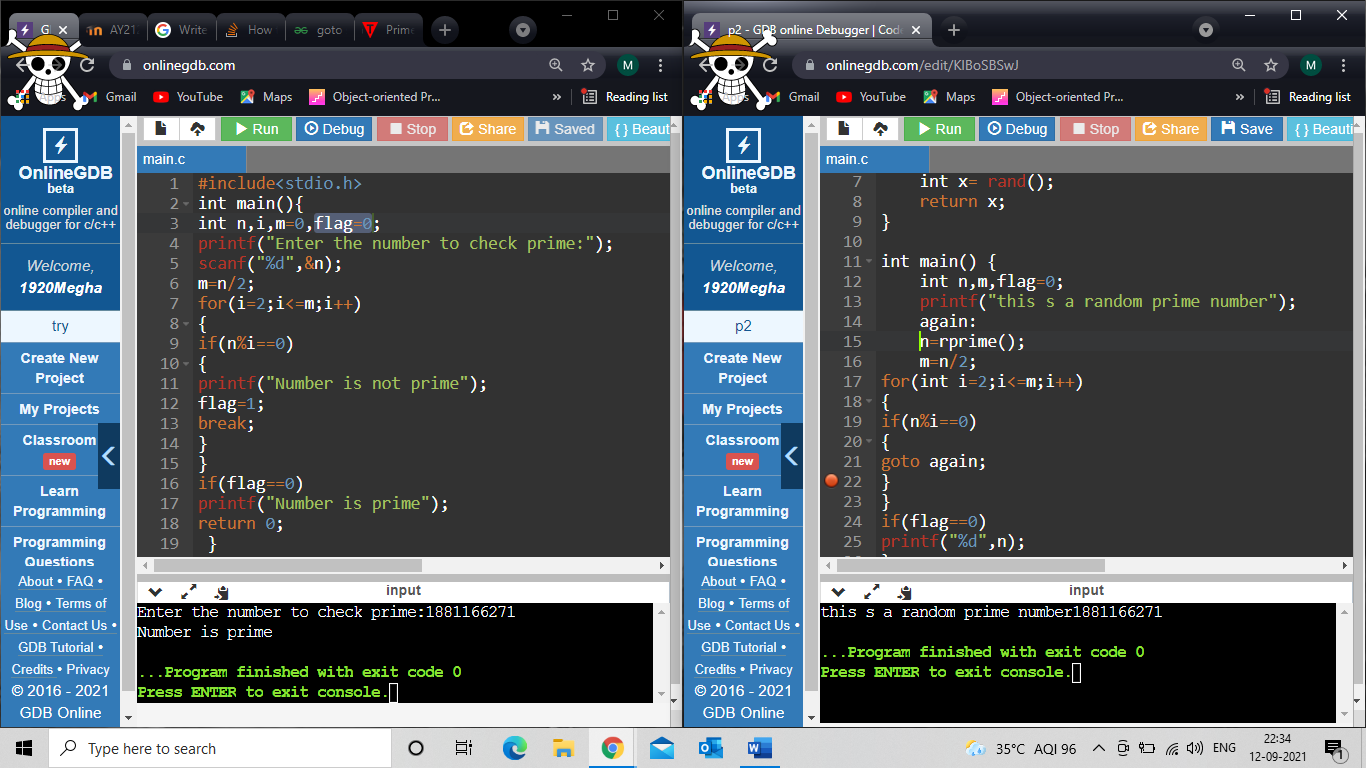
}

if(flag==0)

printf("%d",n);

}





Q.3. Write a C Function to print all natural numbers between start to end.

(Function with arguments and no return value).

INPUT:-

#include <stdio.h>

#include <stdlib.h>

void n(int first, int last)

{

for (int i = first; i <= last; i++)

{

printf("%d ",i);

}

}

int main()

{

int first, last;

printf("Enter first number:");

scanf("%d", &first);

printf("Enter last number:");

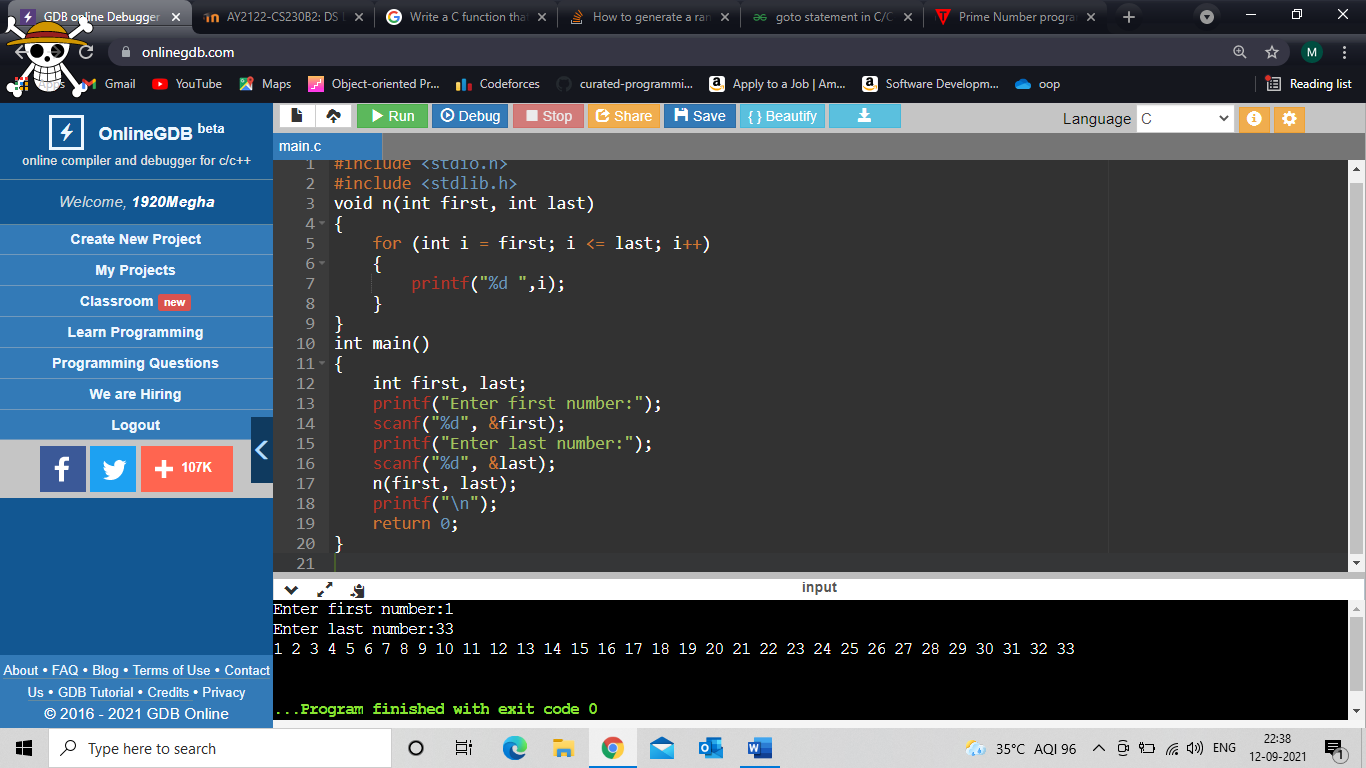
scanf("%d", &last);

n(first, last);

printf("\n");

return 0;

}



Q.4. Write a C program to print all perfect numbers in the given range using

the function. (Function with arguments and a return value).

INPUT:-

#include <stdio.h>

// Function declarations

int isPerfect(int num);

void printPerfect(int start, int end);

int main()

{

int start, end;

// Inputting lower and upper limit to print perfect numbers

printf("Enter lower limit to print perfect numbers: ");

scanf("%d", &start);

printf("Enter upper limit to print perfect numbers: ");

scanf("%d", &end);

printf("All perfect numbers between %d to %d are: \n", start, end);

printPerfect(start, end);

return 0;

}

int isPerfect(int num)

{

int i, sum;

// Finding sum of all proper divisors

sum = 0;

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

{

if(num % i == 0)

{

sum += i;

}

}

/\*

If sum of proper positive divisors equals to given number

then the number is perfect number

\*/

if(sum == num)

return 1;

else

return 0;

}

//Printing all perfect numbers

void printPerfect(int start, int end)

{

// Iterating from start to end

while(start <= end)

{

if(isPerfect(start))

{

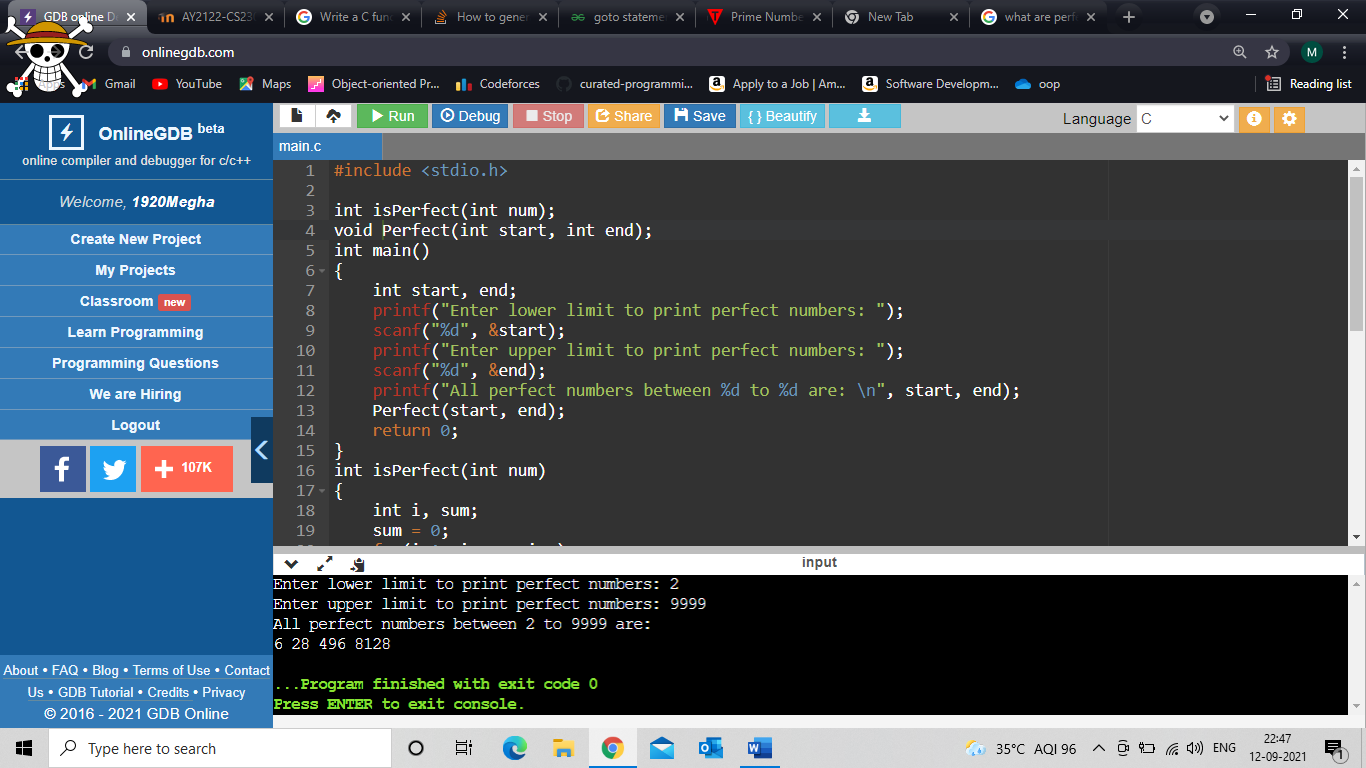
printf("%d ", start);

}

start++;

}

}



Q.5. Write a Program in C to convert decimal number to binary number using

function. (Use all the four types of user defined functions).

i. Function with no arguments and no return value

ii. Function with no arguments and a return value

iii. Function with arguments and no return value

iv. Function with arguments and a return value

#include <stdlib.h>

#include<stdio.h>

int n;

int b()

{

int binary = 0, rem;

int x = n;

for(int i = 1; x != 0; i = i \* 10)

{

rem = x % 2;

x /= 2;

binary += rem \* i;

}

printf("1)\tFunction with no argument and no return value:\n\tBinary number:%d\n", binary);

}

int b2()

{

int binary = 0, rem;

int x = n;

for(int i = 1; x != 0; i = i \* 10)

{

rem = x % 2;

x /= 2;

binary += rem \* i;

}

return binary;

}

int b3(int n)

{

int binary = 0, rem;

for(int i = 1; n != 0; i = i \* 10)

{

rem = n % 2;

n /= 2;

binary += rem \* i;

}

printf("3)\t Function with argument and no return value\n\tBinary number:%d\n", binary);

}

int b4(int n)

{

int binary = 0, rem;

for(int i = 1; n != 0; i = i \* 10)

{

rem = n % 2;

n /= 2;

binary += rem \* i;

}

return binary;

}

int main()

{

printf("Enter a decimal number to convert into binar\n");

printf("enter number for decimal to binary conv\n");

scanf("%d", &n);

b();

printf("2)\tFunction with no argument and a return value:\n\tBinary number:%d\n", b2());

b3(n);

printf("4)\tFunction with argument and a return value:\n\tBinary number:%d\n", b4(n));

}

