***Data Structure and Algorithm   
Practical File***

***By- Himakshi kohli  
2010990293***

***Question 1***

Write a program to swap two numbers using call by value and call by reference.

***Solution-***

***Call by value***

#include<iostream>

using namespace std;

void callByValue(int a, int b){

int temp= a;

a=b;

b=temp;

cout<<"After calling by value:"<<endl;

cout<<" a = "<<a<<", b = "<<b<<endl;

}

int main(){

int a,b;

cout<<"Enter the first number:"<<endl;

cin>>a;

cout<<"Enter the second number:"<<endl;

cin>>b;

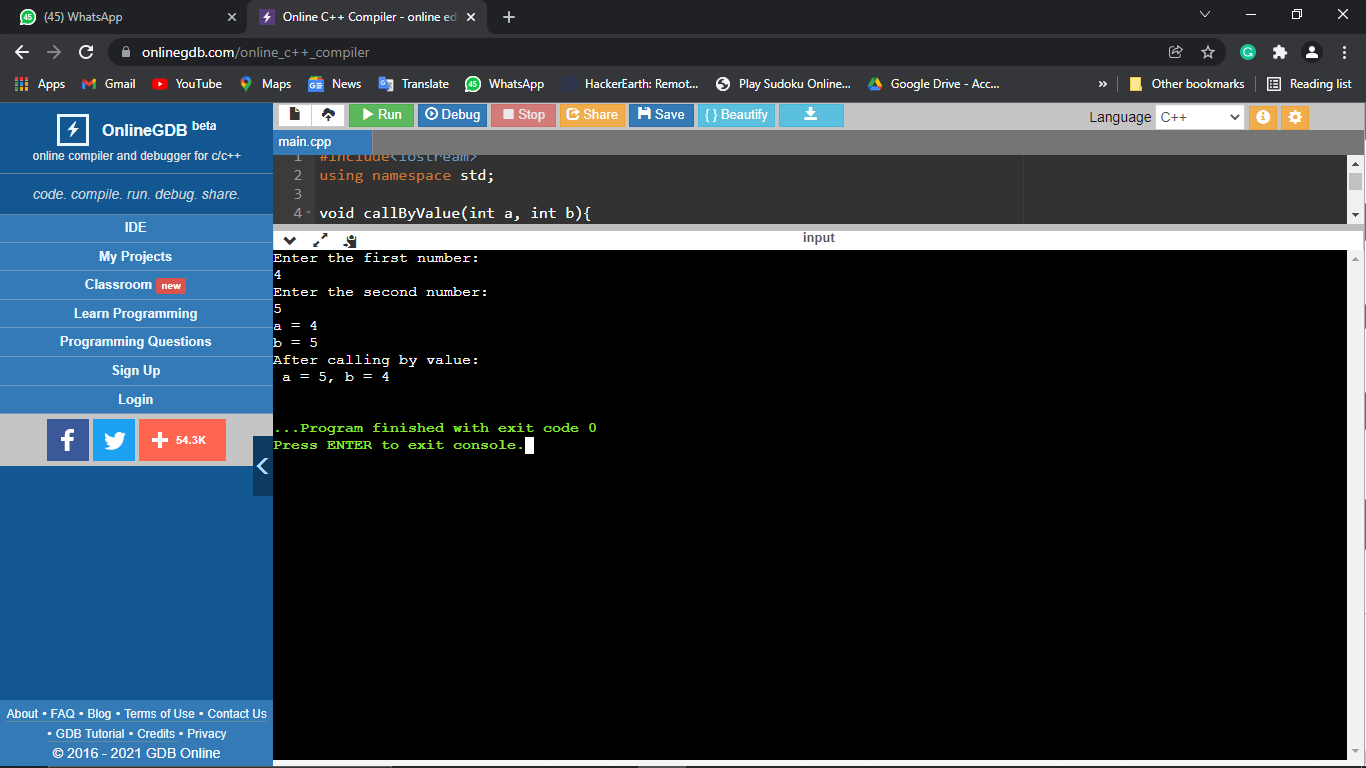
cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

callByValue(a,b);

}

**Output:-**



***Call by Reference***

#include<iostream>

using namespace std;

void callByReference(int \*a, int \*b){

int temp= \*a;

\*a=\*b;

\*b=temp;

cout<<"After calling by reference:"<<endl;

cout<<" a = "<<\*a<<", b = "<<\*b<<endl;

}

int main(){

int a,b;

cout<<"Enter the first number:"<<endl;

cin>>a;

cout<<"Enter the second number:"<<endl;

cin>>b;

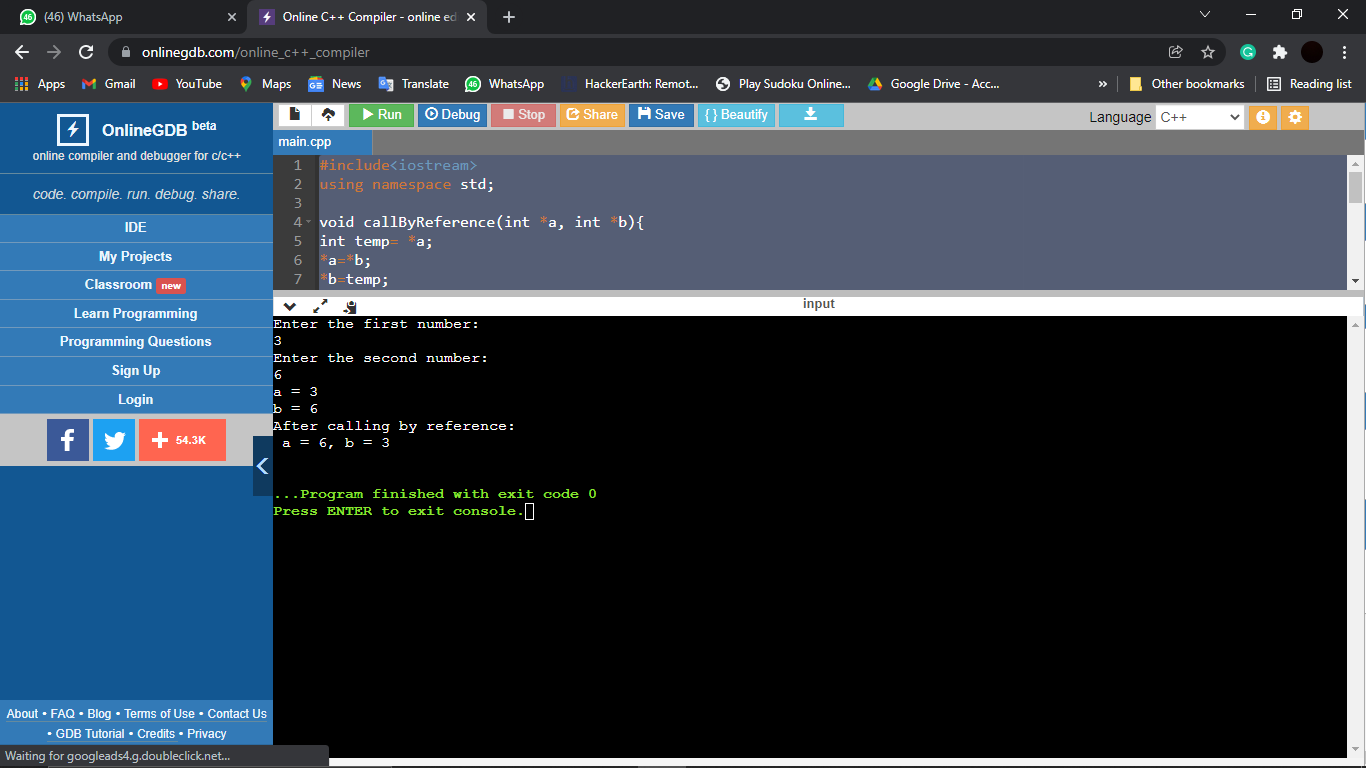
cout<<"a = "<<a<<endl;

cout<<"b = "<<b<<endl;

callByReference(&a,&b);

}

***Output-***



***Question 2***

Write a program to construct a calculator using functions

***Solution-***

#include<iostream>

using namespace std;

int Sum(float a, float b)

{

return a+b;

}

int Sub(int a,int b)

{

return b-a;

}

int Product(int a, int b)

{

return a\*b;

}

int Division(int a, int b)

{

return b/a;

}

int main()

{

int x,y,choice;

cout<<"First number: "<<endl;

cin>>x;

cout<<"Second number: "<<endl;

cin>>y;

cout<<"Enter your choice: "<<endl;

cout<<"1.Addition"<<endl;

cout<<"2.Subtraction"<<endl;

cout<<"3.Multiplication"<<endl;

cout<<"4.Division"<<endl;

cin>>choice;

switch(choice){

case 1:

cout<<"Addition: "<<Sum(x,y);

break;

case 2:

cout<<"Subtraction: "<<Sub(x,y);

break;

case 3:

cout<<"Product: "<<Product(x,y);

break;

case 4:

cout<<"Division: "<<Division(x,y);

default: cout<<"Invalid choice";

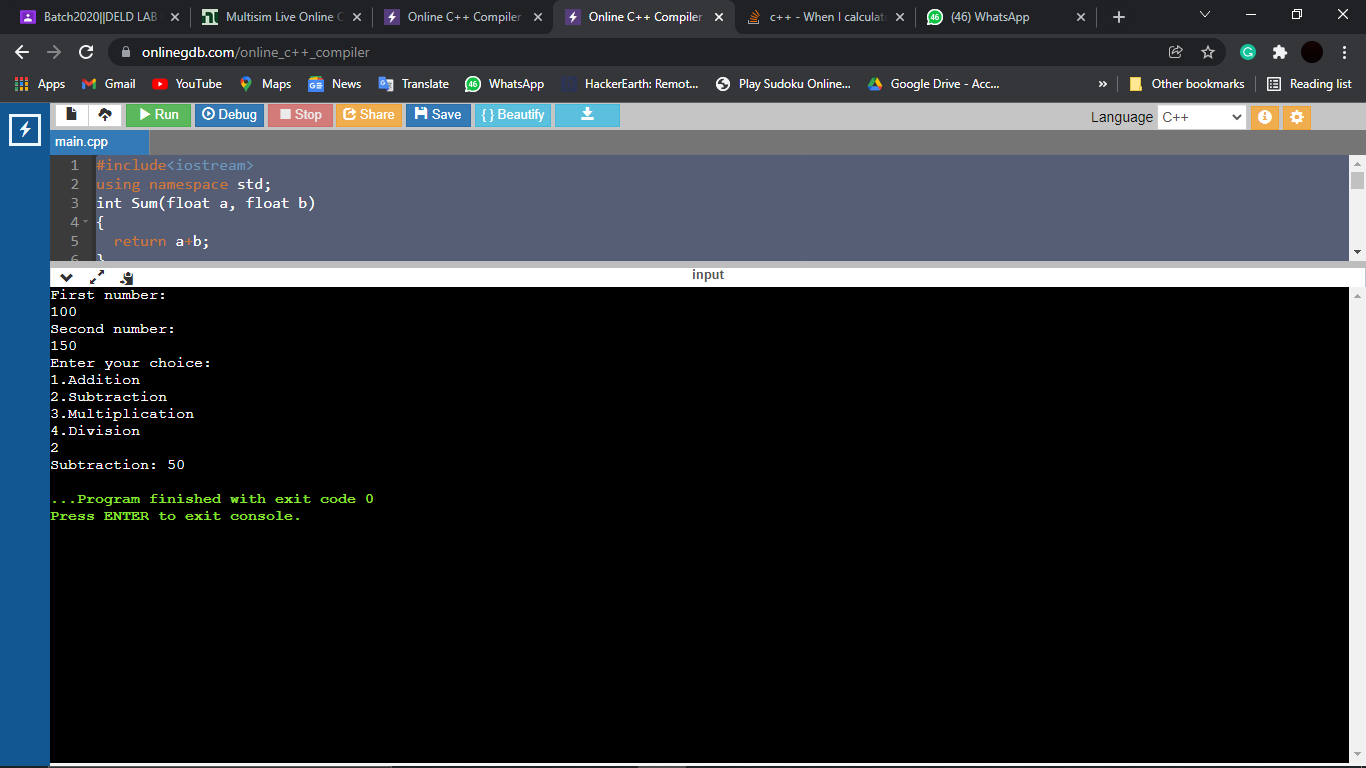
break;

}

return 0;

}

***Output:***



***Question 3***

Write a program to calculate the factorial of a number using functions with return type and arguments.

***Solution-***

#include <iostream>

using namespace std;

int fact(int n){

int i;

unsigned long s=1;

for(i=n;i>=1;i--){

s=s\*i;

}

cout<<s;

return 0;

}

int main()

{

int n;

cout<<"enter number: ";

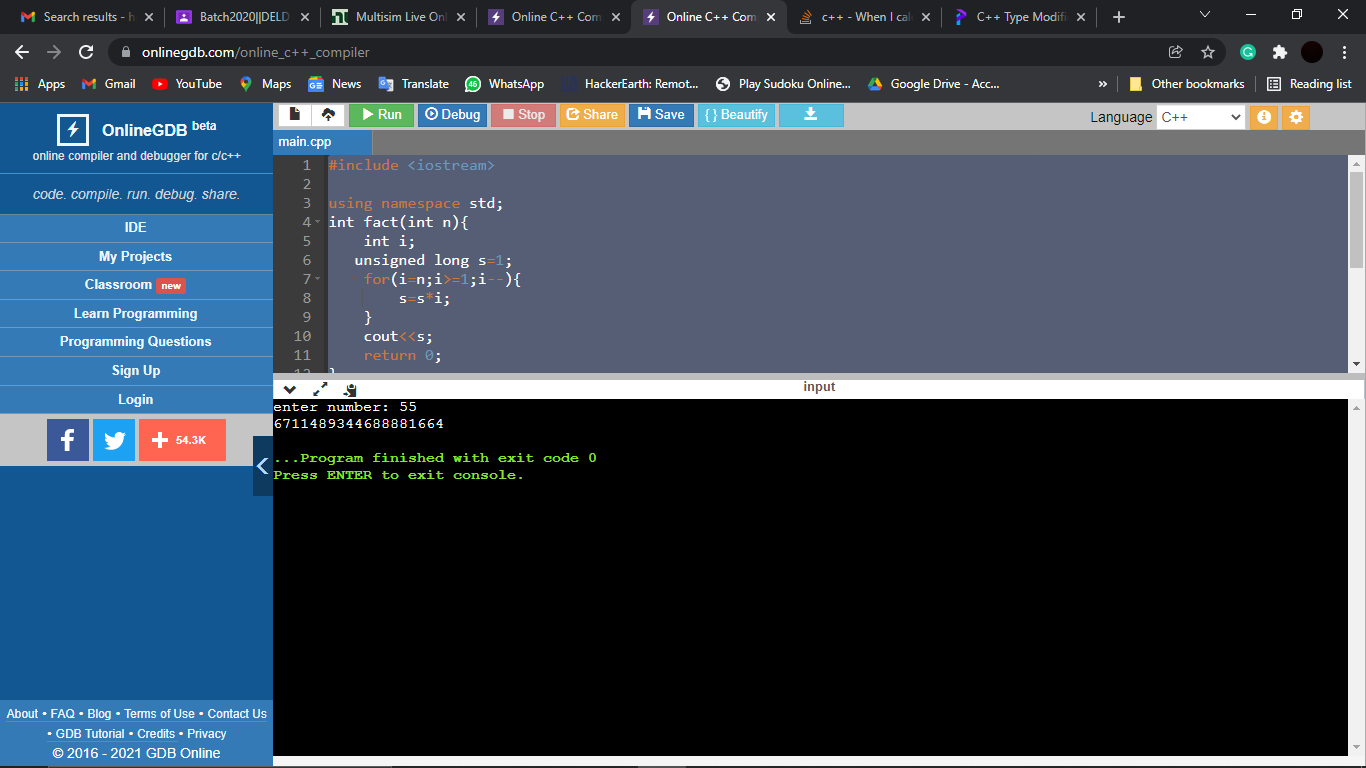
cin>>n;

fact(n);

return 0;

}

Output:



***Question 4***

Write a program to construct a calculator using functions

***Solution –***

#include <iostream>

using namespace std;

int main(){

cout<<"Pointer to pointer : "<<endl;

int n;

int \*ptrn;

int \*\*ptrp;

cout<<"Enter any number : ";

cin>>n;

ptrn = &n;

ptrp = &ptrn;

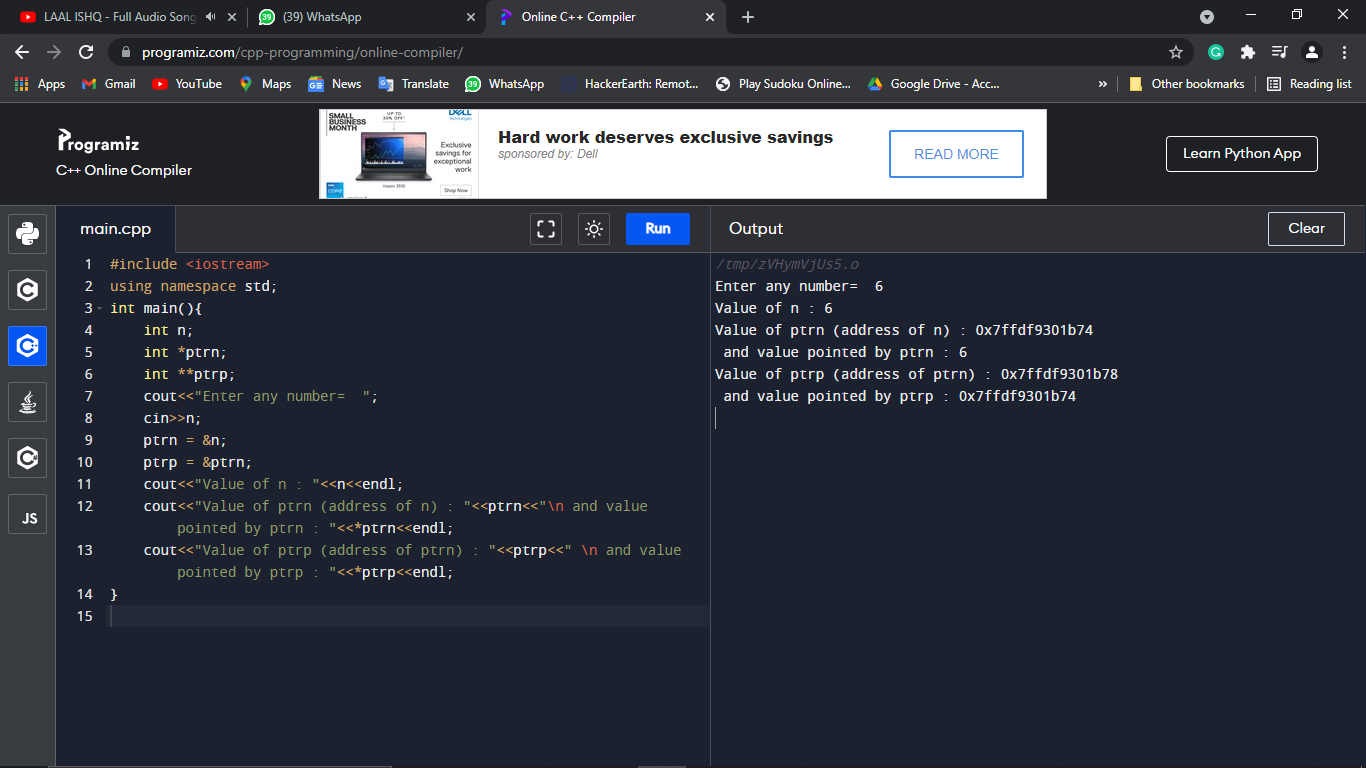
cout<<"Value of n : "<<n<<endl;

cout<<"Value of ptrn (address of n) : "<<ptrn<<" and value pointed by ptrn : "<<\*ptrn<<endl;

cout<<"Value of ptrp (address of ptrn) : "<<ptrp<<" and value pointed by ptrp : "<<\*ptrp<<endl;

}

Output



***Question 5***

Write a program to create a database of a student with student name roll number grades of three subjects using classes

***Solution -***

#include <iostream>

using namespace std;

class Students{

private :

int rollNo;

string studentName;

int pMarks;

int cMarks;

int mMarks;

public:

void setName(string name){

studentName = name;

}

void setRollNo(int n){

if(n<1){

cout<<"Invalid Roll No";

return;

}

else{

rollNo = n;

}

}

void setMarks(int m1, int m2, int m3){

if(m1<0 || m2<0 || m3<0 || m1>100 || m2>100 || m3>100){

cout<<"Invalid Marks";

return;

}

else{

pMarks = m1;

cMarks = m2;

mMarks = m3;

}

}

void displayName(){

cout<<"Name : "<<studentName<<endl;

}

void displayRollNo(){

cout<<"Roll No : "<<rollNo<<endl;

}

void displayGrade(){

int avg = pMarks+cMarks+mMarks/3;

if(avg<40){

cout<<'C'<<endl;

}

else if(avg>=40 && avg<70){

cout<<'B'<<endl;

}

else{

cout<<'A'<<endl;

}

}

};

int main(){

cout<<"Classes and Objects : "<<endl;

Students s1,s2,s3;

int n1,n2,n3,a1,a2,a3,b1,b2,b3,c1,c2,c3;

string name1, name2, name3;

cout<<"Enter the name of student 1 : "<<endl;

cin>>name1;

cout<<"Enter the roll no of student 1 : "<<endl;

cin>>n1;

cout<<"Enter the marks of student 1 : "<<endl;

cin>>a1>>b1>>c1;

cout<<"Enter the name of student 2 : "<<endl;

cin>>name2;

cout<<"Enter the roll no of student 2 : "<<endl;

cin>>n2;

cout<<"Enter the marks of student 2 : "<<endl;

cin>>a2>>b2>>c2;

cout<<"Enter the name of student 3 : "<<endl;

cin>>name3;

cout<<"Enter the roll no of student 3 : "<<endl;

cin>>n3;

cout<<"Enter the marks of student 3 : "<<endl;

cin>>a3>>b3>>c3;

s1.setName(name1);

s1.setRollNo(n1);

s1.setMarks(a1,b1,c1);

s2.setName(name2);

s2.setRollNo(n2);

s2.setMarks(a2,b2,c2);

s3.setName(name3);

s3.setRollNo(n3);

s3.setMarks(a3,b3,c3);

cout<<"Info of student 1 : "<<endl;

s1.displayName();

s1.displayRollNo();

s1.displayGrade();

cout<<"Info of student 2 : "<<endl;

s2.displayName();

s2.displayRollNo();

s2.displayGrade();

cout<<"Info of student 3 : "<<endl;

s3.displayName();

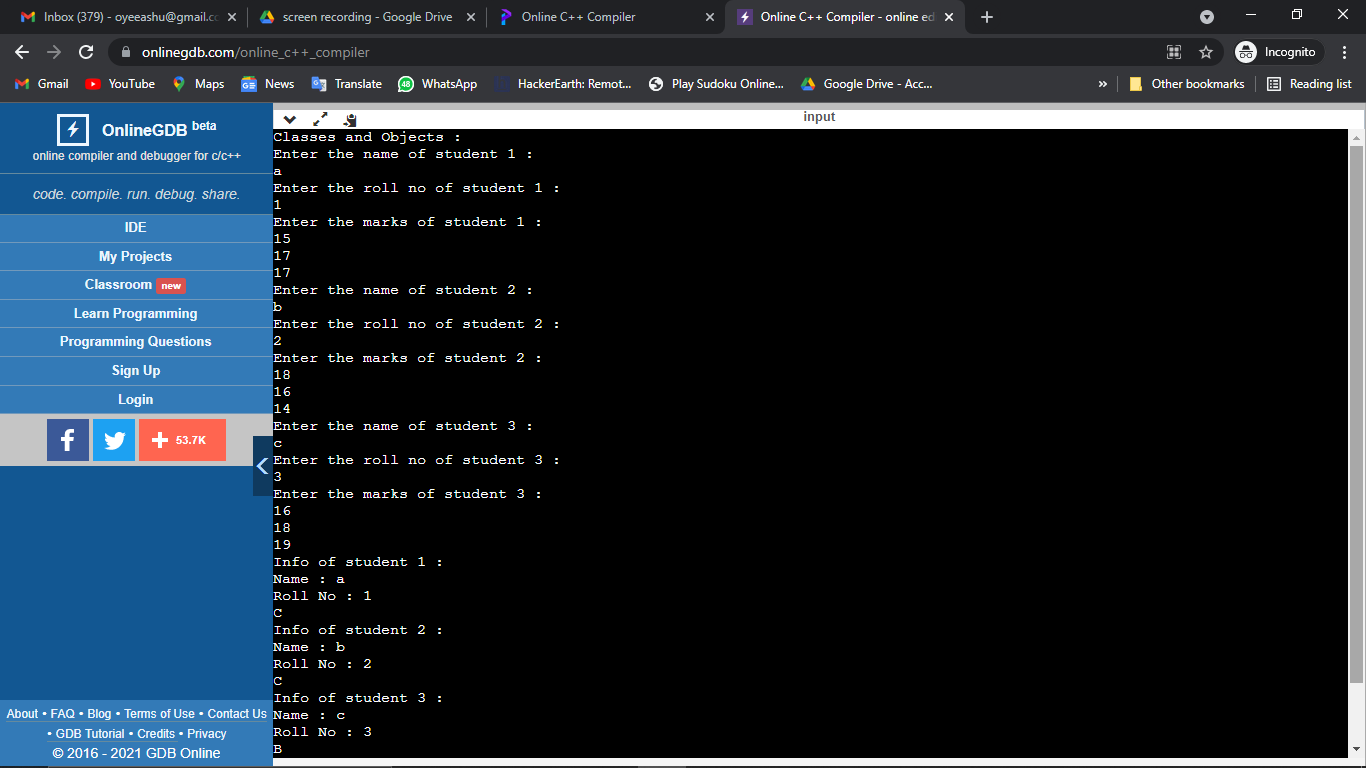
s3.displayRollNo();

s3.displayGrade();

return 0;

}

Output:



***Question 6***

Write a program to demonstrate the use of copy constructor?

***Solution-***

#include <iostream>

using namespace std;

class Wall {

private:

double length;

public:

Wall() {

float length;

cout << "Creating a wall." << endl;

cin>>length;

cout << "Length = " << length << endl;

}

};

int main() {

Wall wall1;

return 0;

}

***Question 7***

Write a program for new and delete operator use.

***Solution-***

#include <iostream>

using namespace std;

int main()

{

int \*p1, \*p2, mul;

p1=new int;

p2= new int;

cout<<"Enter first value: ";

cin>>\*p1;

cout<<"Enter second value: ";

cin>>\*p2;

mul = \*p1 \* \*p2;

cout<<"Product: "<<mul<<endl;

delete p1;

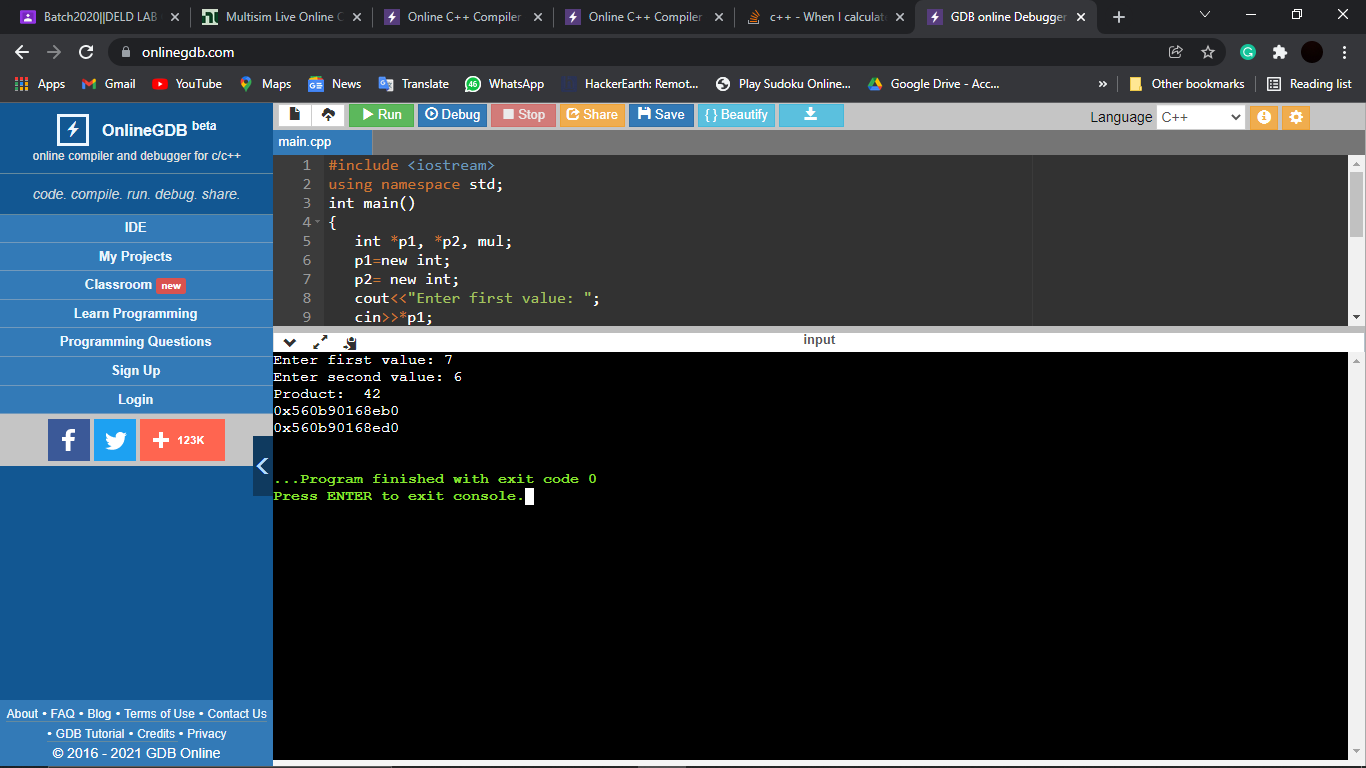
delete p2;

cout<<p1<<endl;

cout<<p2<<endl;

}

***Output***:



***Question 8***

Write a program to draw the pattern.

A

AB

ABC

ABCD

ABCDE

***Solution-***

#include <iostream>

using namespace std;

int main(){

int n;

cout<<"Enter the number of rows: ";

cin>>n;

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

char k='A';

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

cout<<k;

k++;

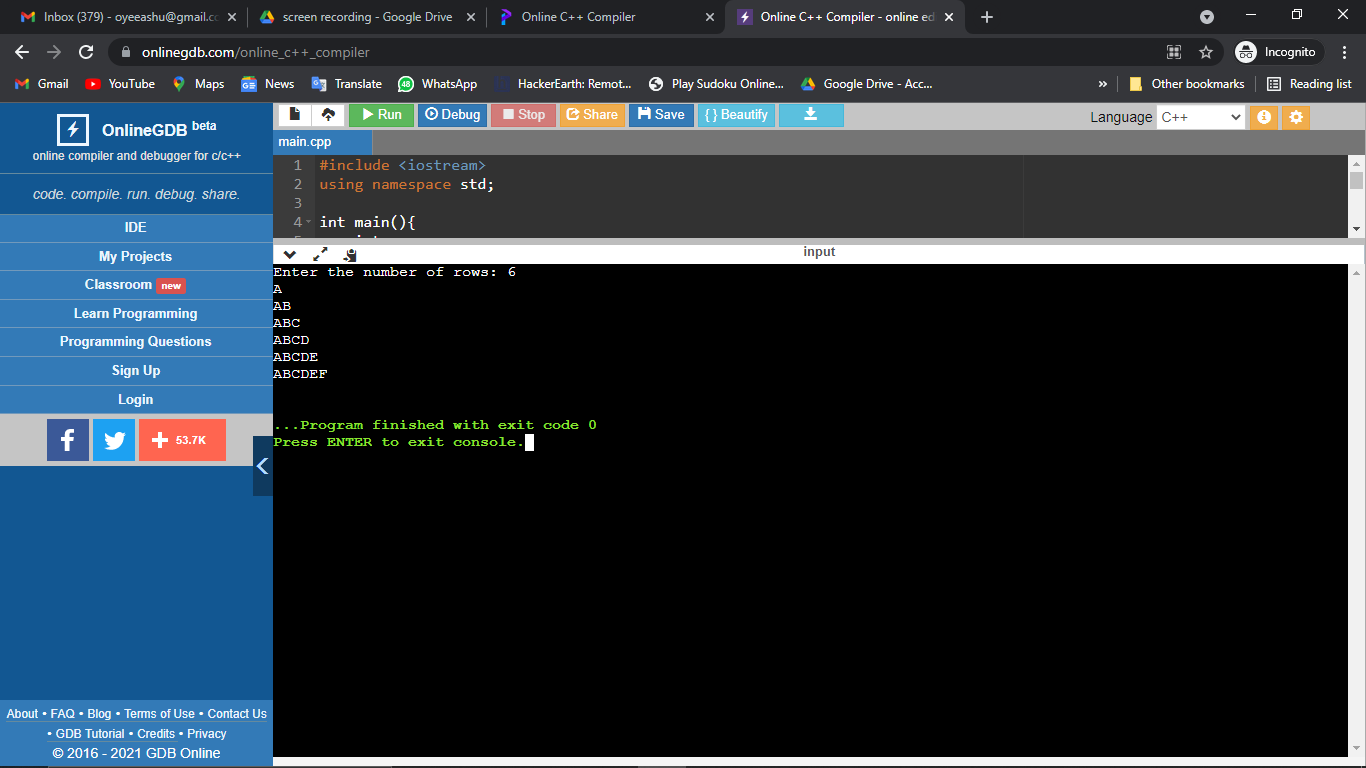
}

cout<<endl;

}

}

Output:



***Question 9***

Print the pattern

1

12

123

1234

**Solution-**

#include <iostream>

using namespace std;

int main(){

int n;

cout<<"Enter the number of rows: ";

cin>>n;

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

int k=1;

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

cout<<k;

k++;

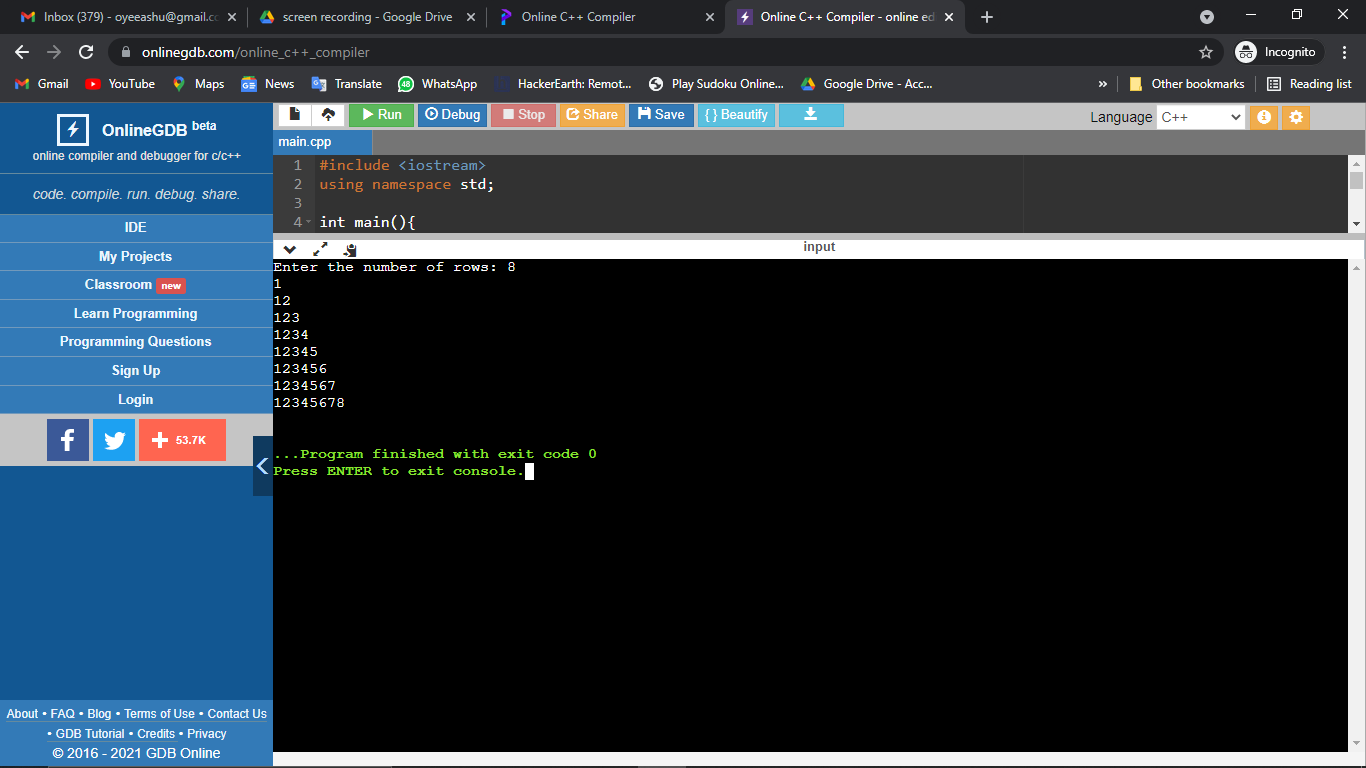
}

cout<<endl;

}

}

Output:



***Question 10***

Print the pattern

1

2 3

4 5 6

7 8 9 10

***Solution-***

#include <iostream>

using namespace std;

int main(){

cout<<"Pattern 3 : "<<endl;

int n,k=1;

cout<<"Enter the number of rows: ";

cin>>n;

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

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

cout<<k<<" ";

k++;

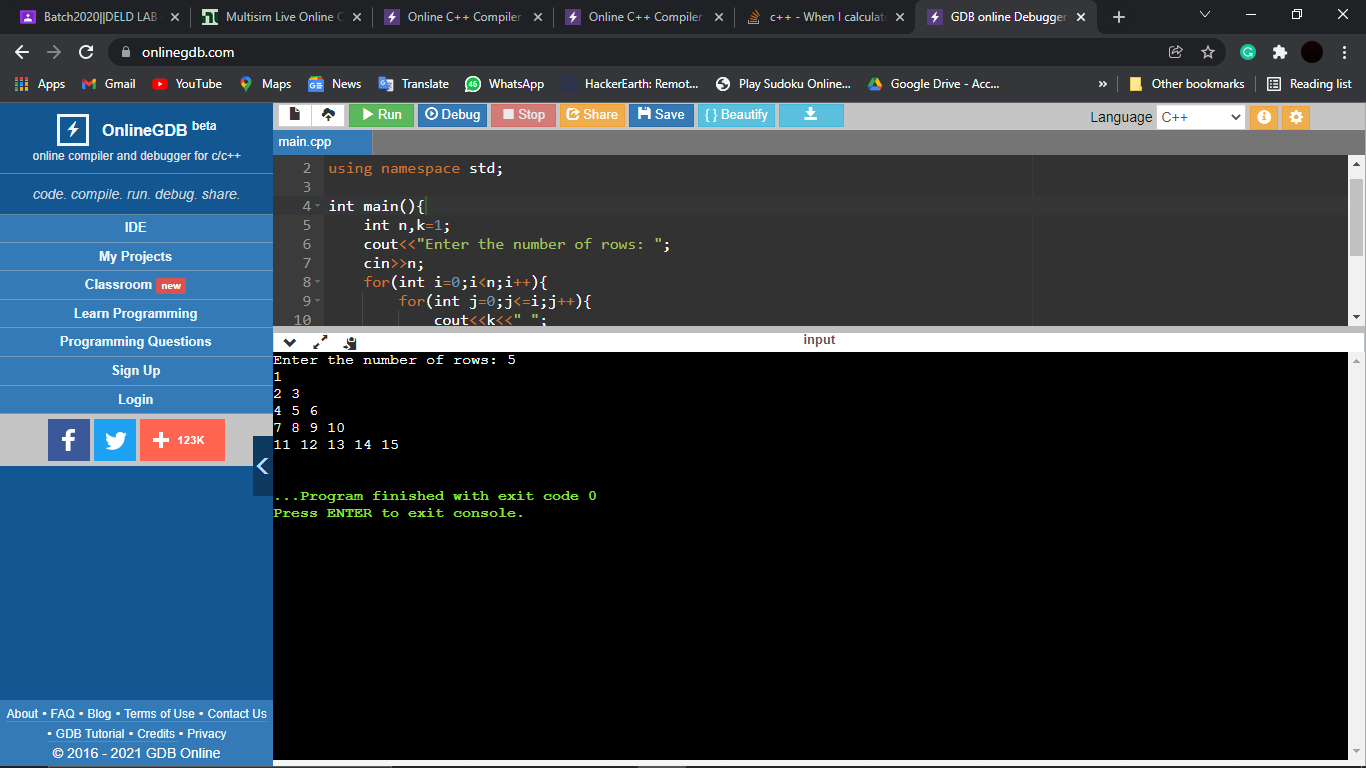
}

cout<<endl;

}

}

***Output***



***Question 11***

Write a program to check whether a given is palindrome or not?

***Solution-***

#include<iostream>

using namespace std;

void palindrome(int num){

int remainder;

int rev = 0;

int temp = num;

while(temp > 0){

remainder = temp%10;

rev = (rev\*10)+remainder;

temp = temp/10;

}

if(rev==num)

cout<<"It is a Palindrome Number" << endl;

else

cout<<"It is not a Palindrome Number" << endl;

}

int main()

{

int n;

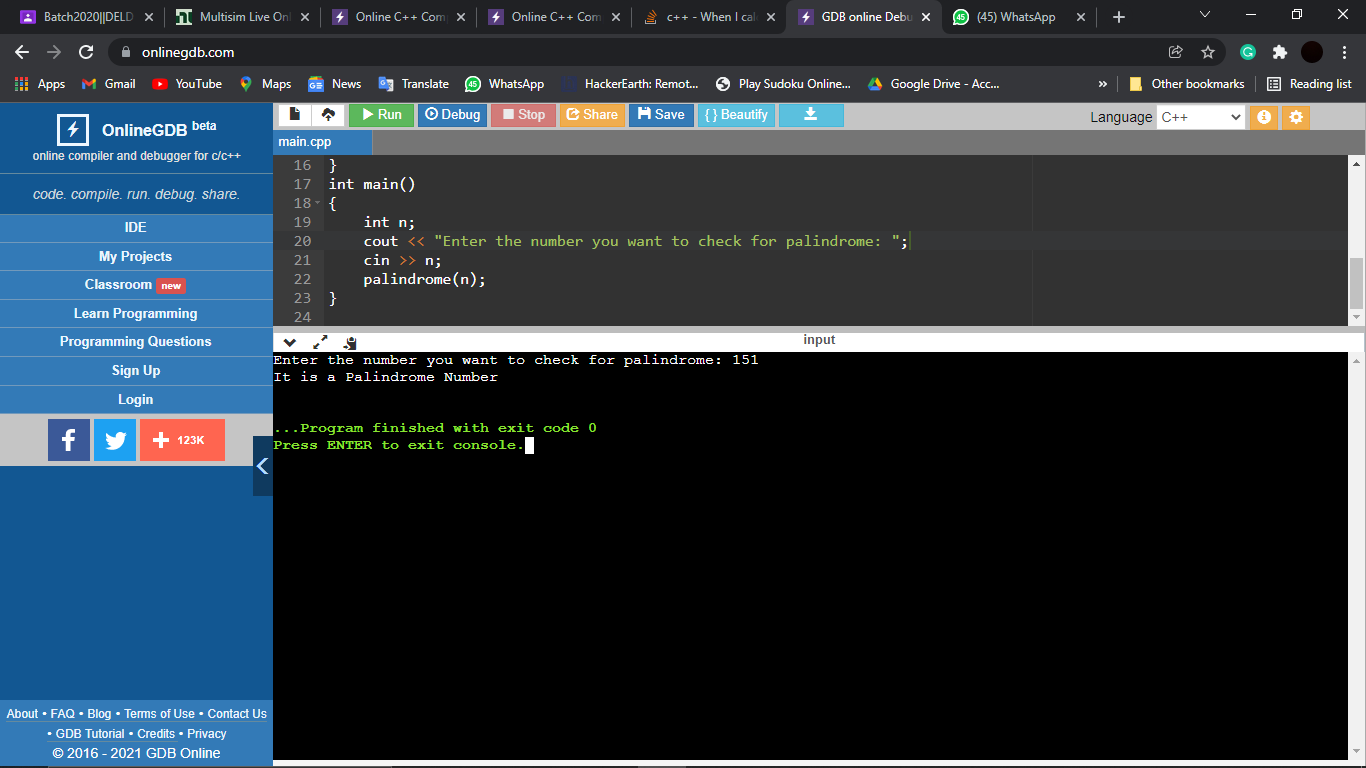
cout << "Enter the number you want to check for palindrome : ";

cin >> n;

palindrome(n);

}

Output:



***Question12***

Write a program to find the greatest among 3 numbers using functions?

***Solution-***

#include <iostream>

using namespace std;

int greatest(int n1,int n2,int n3){

if(n1>=n2 && n1>=n2){

return n1 ;

}

else if(n2>=n1 && n2>=n3){

return n2;

}

else{

return n3;

}

}

int main(){

int x,y,z;

cout << "Enter the first number: " << endl;

cin >> x;

cout << "Enter the Second number: " << endl;

cin >> y;

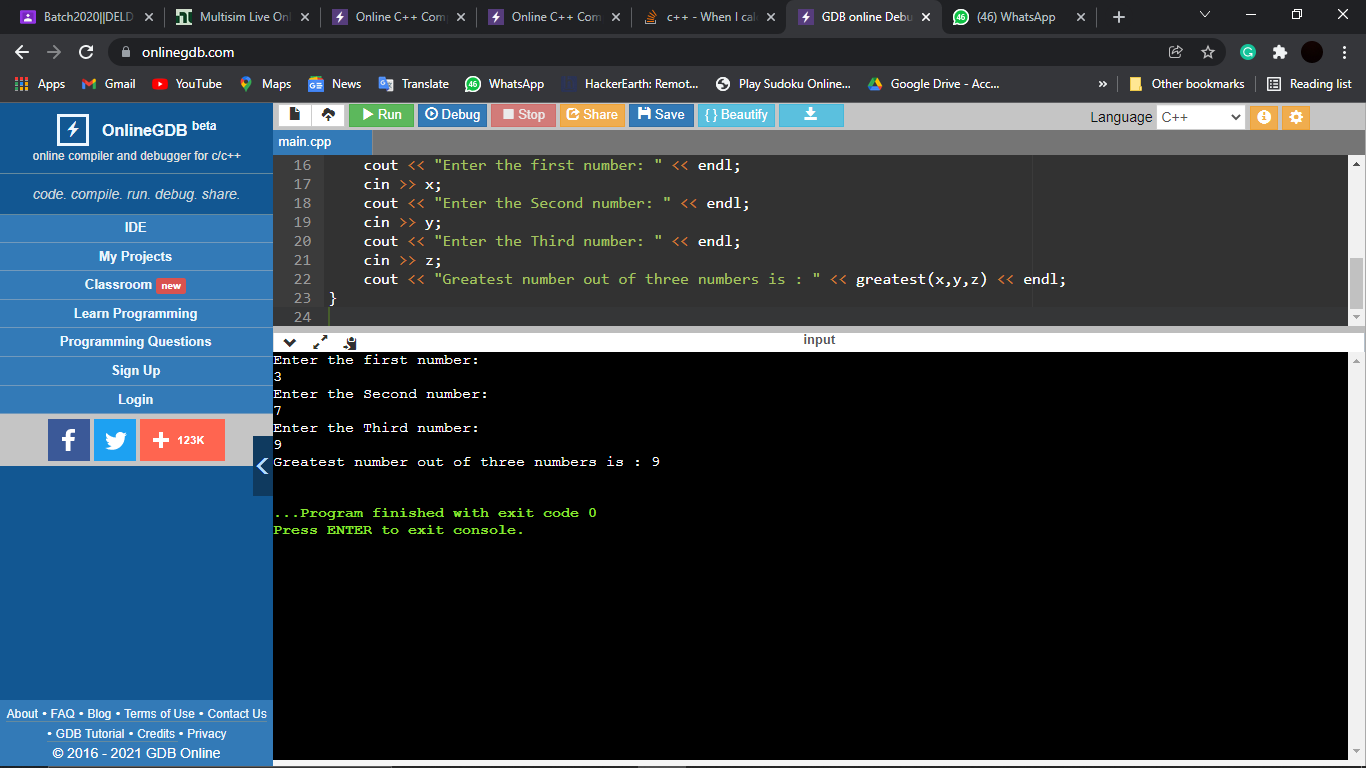
cout << "Enter the Third number: " << endl;

cin >> z;

cout << "Greatest number out of three numbers is : " << greatest(x,y,z) << endl;

}

***Output:***



***Question 13***

write a program to check given number is even or odd?

***Solution-***

#include <iostream>

using namespace std;

int main(){

int n;

cout<<"Enter the number to check even or odd: ";

cin>>n;

if(n%2==0){

cout<<n<< " is a even number";

}

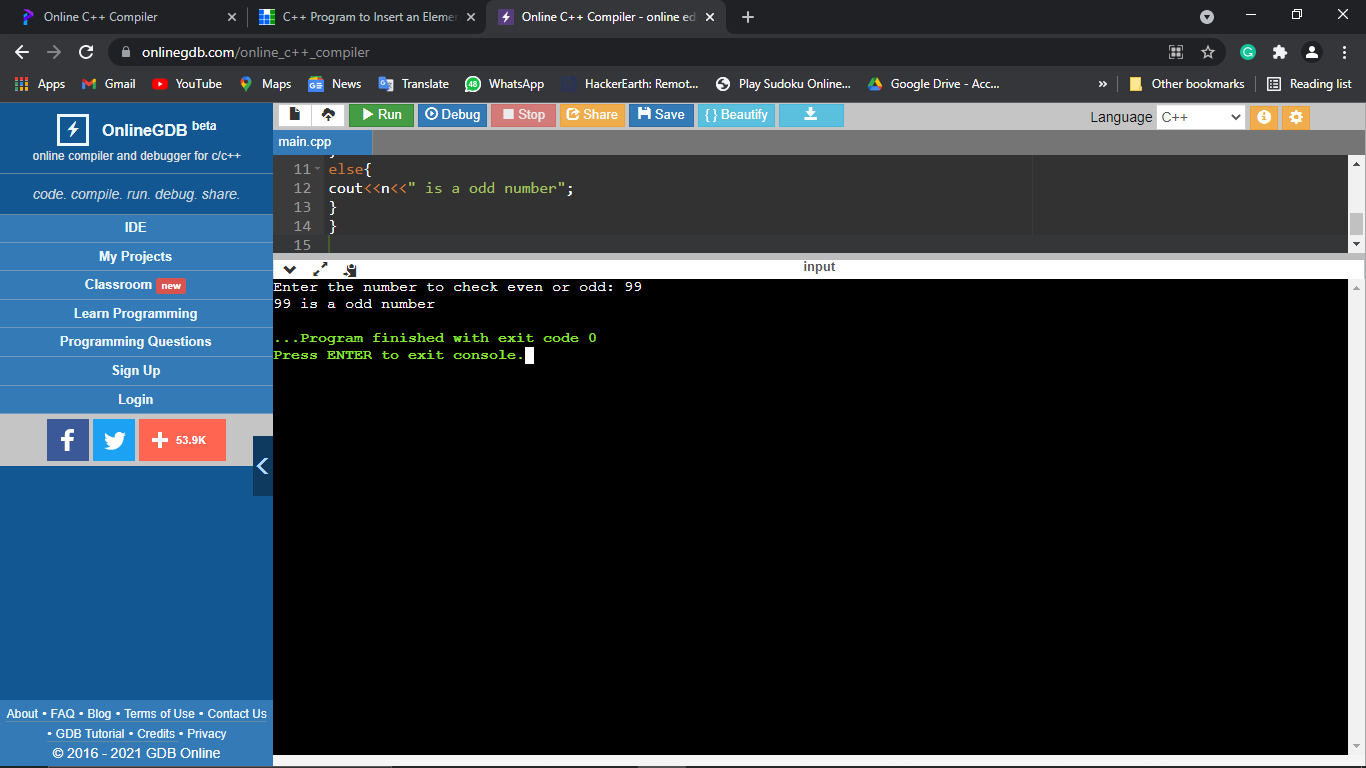
else{

cout<<n<<" is a odd number";

}

}

***Output:***



***Question 14***

Write a program to show the use of function overloading with the name of function sum

***Solution-***

#include <iostream>

using namespace std;

int sum(int a,int b){

return a+b;

}

int sum(int a,int b,int c){

return a+b+c;

}

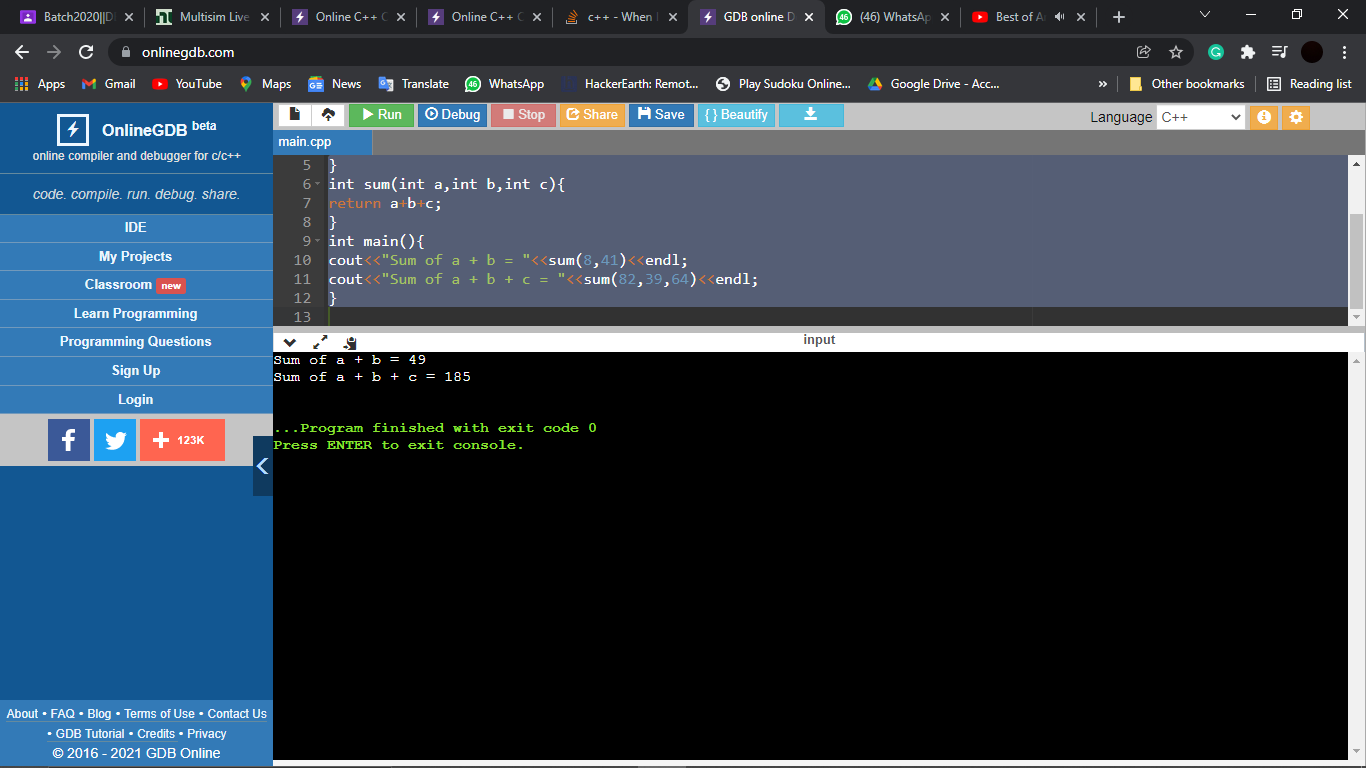
int main(){

cout<<"Sum of a + b = "<<sum(8,41)<<endl;

cout<<"Sum of a + b + c = "<<sum(82,39,64)<<endl;

}

***Output-***



***Question15***

Write a program to convert binary to decimal?

***Solution-***

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

long i,n,num=0,d;

cout<<"Enter any Binary number:";

cin>>n;

cout<<"\nThe Decimal conversion of "<<n<<" is ";

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

{

d=n%10;

num=(d)\*(pow(2,i))+num;

n=n/10;

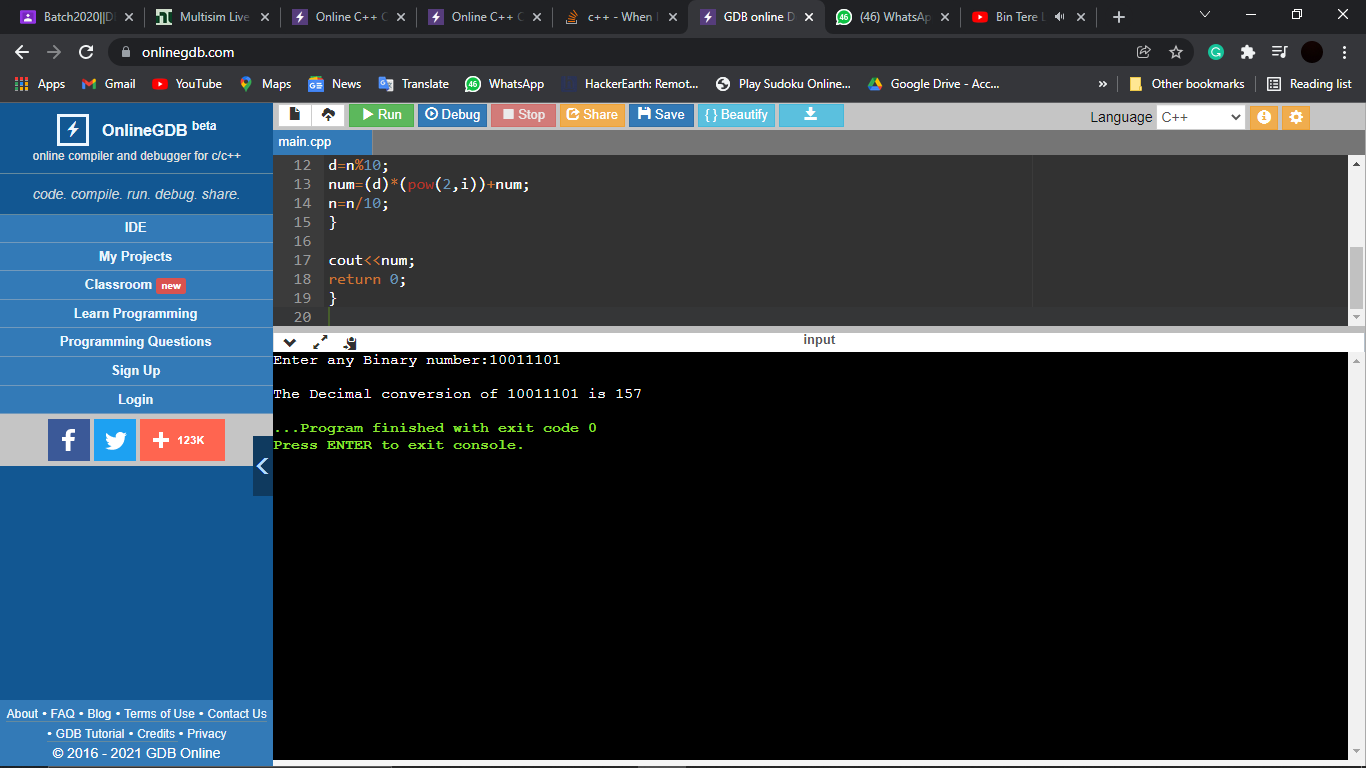
}

cout<<num;

return 0;

}

***Output-***



***Question16***

Write a program to demonstrate the use of constructor and destructor?

***Solution-***

#include <iostream>

using namespace std;

class IntegerSum {

private:

int a,b;

public:

IntegerSum(int x, int y ) { // parametric constructor

a = x;

b = y;

}

~ IntegerSum () { // destructor

cout << "Object is destroyed successfully" << endl;

}

int display () {

return a + b;

}

};

int main()

{

int x,y;

cout<<"Enter number 1: ";

cin>>x;

cout<<"Enter number 2: ";

cin>>y;

IntegerSum sum1(x,y);

IntegerSum sum2(sum1);

IntegerSum sum3 = sum2;

cout << sum1.display() << endl;

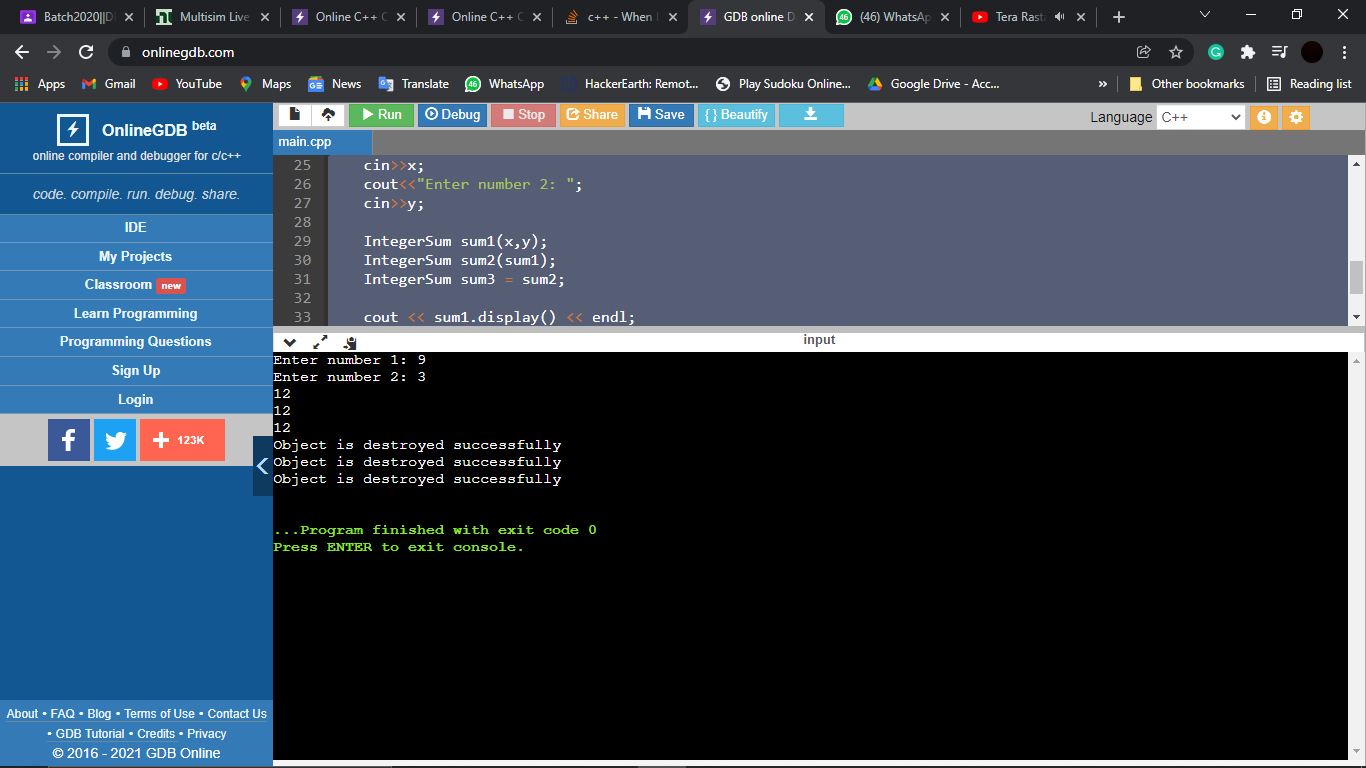
cout << sum2.display() << endl;

cout << sum3.display() << endl;

return 0;

}

***Output-***



***Question17***

Write a program to insert an element in an array?

***Solution-***

#include <iostream>

using namespace std;

int main(){

int n,element,position;

cout << “Length of array : ";

cin >> n;

int arr[n] = {};

cout << "Enter the elements of array : ";

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

{

cin >> arr[i];

}

cout << "Enter the position you want to insert : ";

cin >> position;

cout << "Enter the element you want to add at position : ";

cin >> element;

for(int i = n;i>=position;i--)

{

arr[i] = arr[i-1];

}

arr[position] = element;

n++;

cout << "The new array is ";

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

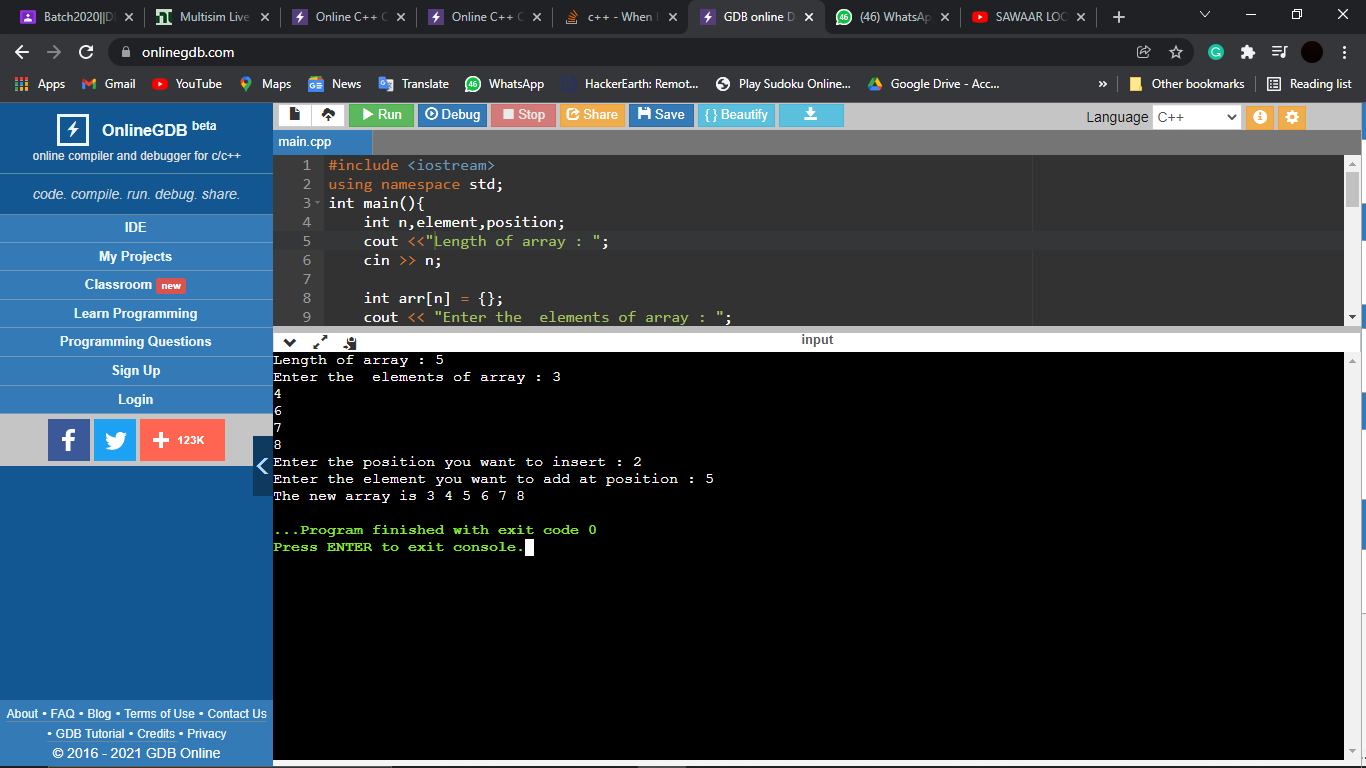
{

cout << arr[i] << " ";

}

}

***Output-***



***Question18***

Write a program to find the value and delete from a given array?

***Solution-***

#include <iostream>

using namespace std;

int main(){

int size,element,del,count=0;

cout << “Length of array : ";

cin >> size;

int arr[size] = {};

cout << "Enter the elements of array : ";

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

{

cin >> arr[i];

}

cout << "Enter the element to be delete: " ;

cin >> del;

cout << "The array before deleting : ";

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

{

cout << arr[i] << " ";

}

cout << endl;

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

{

if(arr[i]==del)

{

for(int j=i; j<(size-1); j++)

{

arr[j]=arr[j+1];

}

count++;

break;

}

}

if(count==0)

{

cout<<"Element not found.\n";

}

else

{

cout<<"Element deleted successfully.\n";

cout << "The new array: ";

for(int i = 0;i<size-1;i++)

{

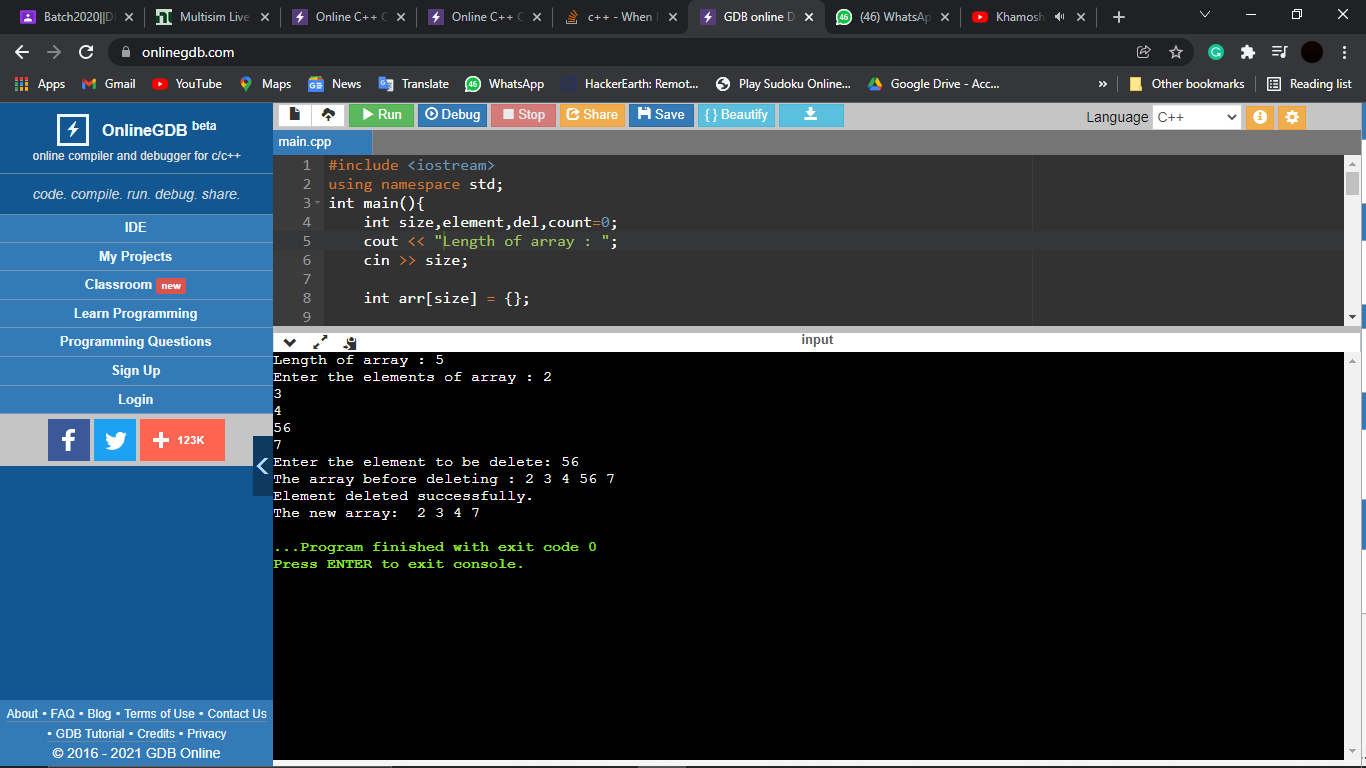
cout << arr[i] << " ";

}

}

}

***Output-***



***Question 19***

Write a program to implement binary search in a given array?

***Solution-***

#include <iostream>

using namespace std;

int main()

{

int length, element, start, end, mid = -1;

cout << "Enter the size of Array: ";

cin >> length;

int arr[length];

cout << "Enter the elements of the array: ";

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

cin >> arr[i];

}

cout << "Enter the element to be searched: ";

cin >> element;

cout << "Element found at index : " << endl;

start = 0, end = length - 1;

while (start <= end) {

mid = (start+end)/2;

if (arr[mid] == element) {

cout << mid;

break;

}

else if (arr[mid] < element)

start = mid + 1;

else

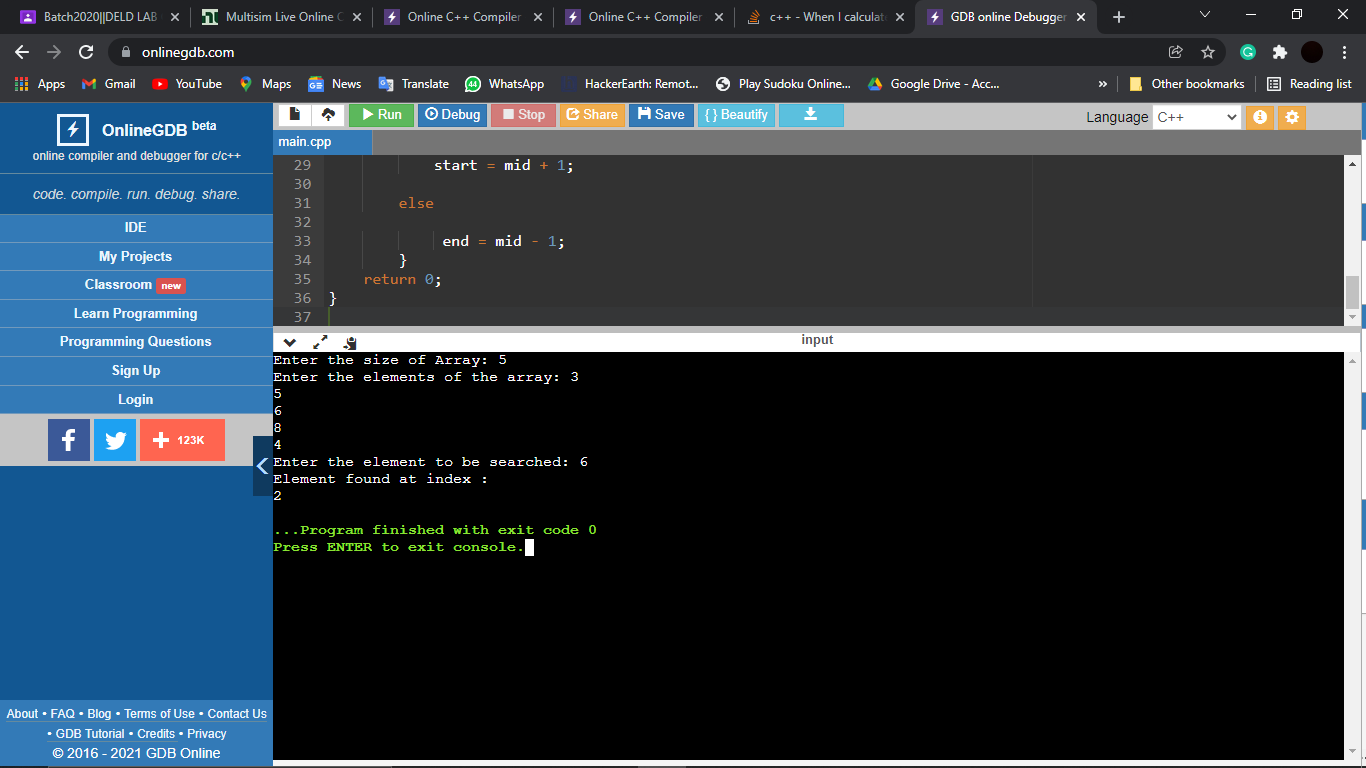
end = mid - 1;

}

return 0;

}

***Output:***



***Question 20***

Write a program to implement the following operation of singly linked list?

1)Insert an element at beginning

***Solution-***

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

void addAtStart(Node\* &head, int val) {

Node\* n = new Node();

n -> data = val;

n -> next = head; // n de next ch head da address

head = n;

}

void display(Node\* n) {

while (n != NULL) {

cout << n -> data << " -> ";

n = n -> next;

}

cout << "NULL" << endl;

}

int main()

{

Node\* head = NULL;

int i;

char ch = 'y';

while (ch == 'y' || ch == 'Y') {

cout << "Enter number to be inserted: ";

cin >> i;

addAtStart(head,i);

cout << "Want to add more numbers(y/Y): ";

cin >> ch;

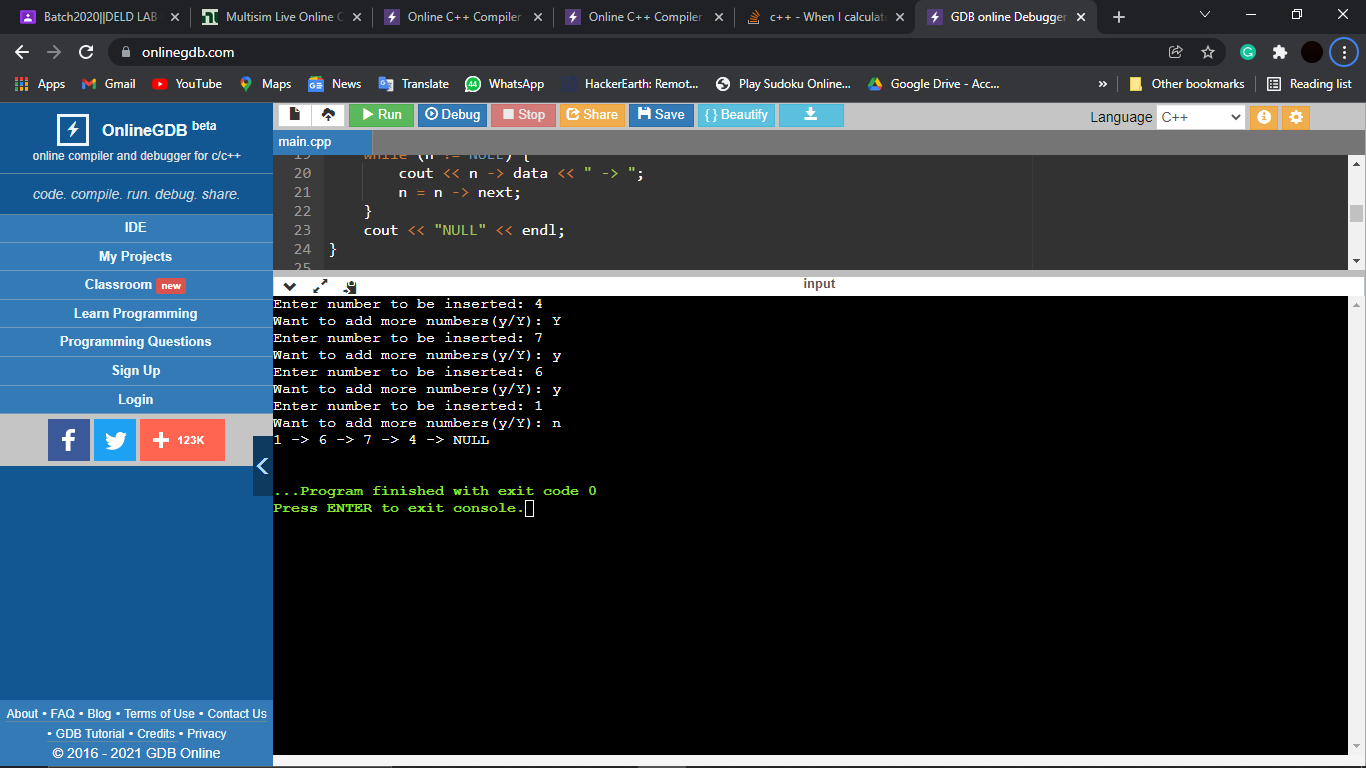
}

display(head);

return 0;

}

***Output-***



(B). Insert at given location

***Solution-***

#include <iostream>

using namespace std;

class Node{

public:

int data;

Node\* next;

};

void insertBetween(Node\* add\_after, int data)

{

if (add\_after == NULL)

{

cout << "the given previous node cannot be NULL";

return;

}

Node\* temp = new Node();

temp->data = data;

temp->next = add\_after->next;

add\_after->next = temp;

}

void display(Node\* head)

{

while(head!=NULL)

{

cout<<head->data<<" -> ";

head = head-> next;

}

}

int main(){

Node\* head=new Node();

Node\* second=new Node();

Node\* third=new Node();

Node\* fourth=new Node();

head->data=1;

head->next=second;

second->data=2;

second->next=third;

third->data=3;

third->next=fourth;

fourth->data=4;

fourth->next=NULL;

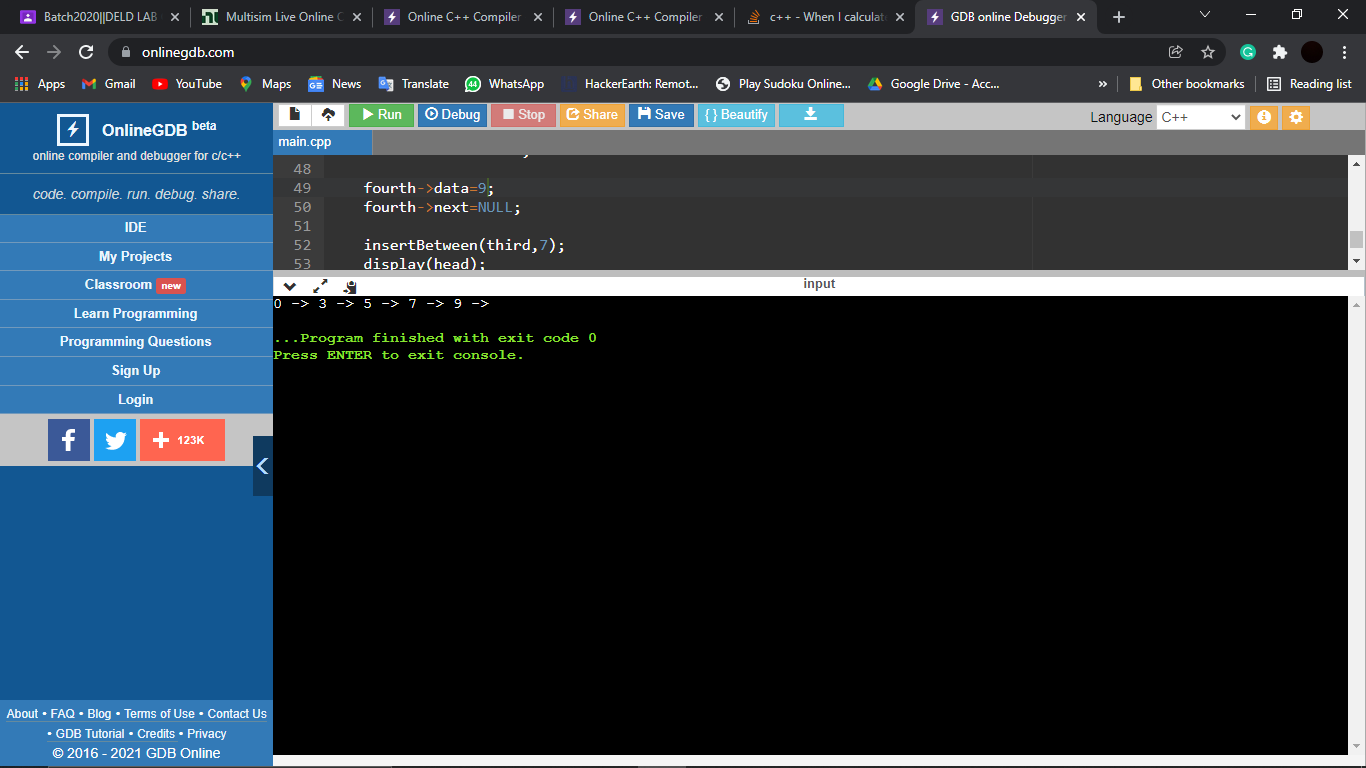
insertBetween(third,5);

display(head);

return 0;

}

***Output-***



(C)Insert an element at end of list

***Solution-***

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

void addAtEnd(Node\* &head, int val) {

Node\* n = new Node();

n -> data = val;

n -> next = NULL;

if(head == NULL) {

head = n;

return;

}

Node\* temp = head;

while (temp -> next != NULL) {

temp = temp -> next;

}

temp -> next = n;

}

void display(Node\* n) {

while (n != NULL) {

cout << n -> data << " -> ";

n = n -> next;

}

cout << "NULL" << endl;

}

int main()

{

Node\* head = NULL;

int i;

char ch = 'y';

while (ch == 'y' || ch == 'Y') {

cout << "Enter number to be inserted: ";

cin >> i;

addAtEnd(head,i);

cout << "Want to add more numbers(y/Y): ";

cin >> ch;

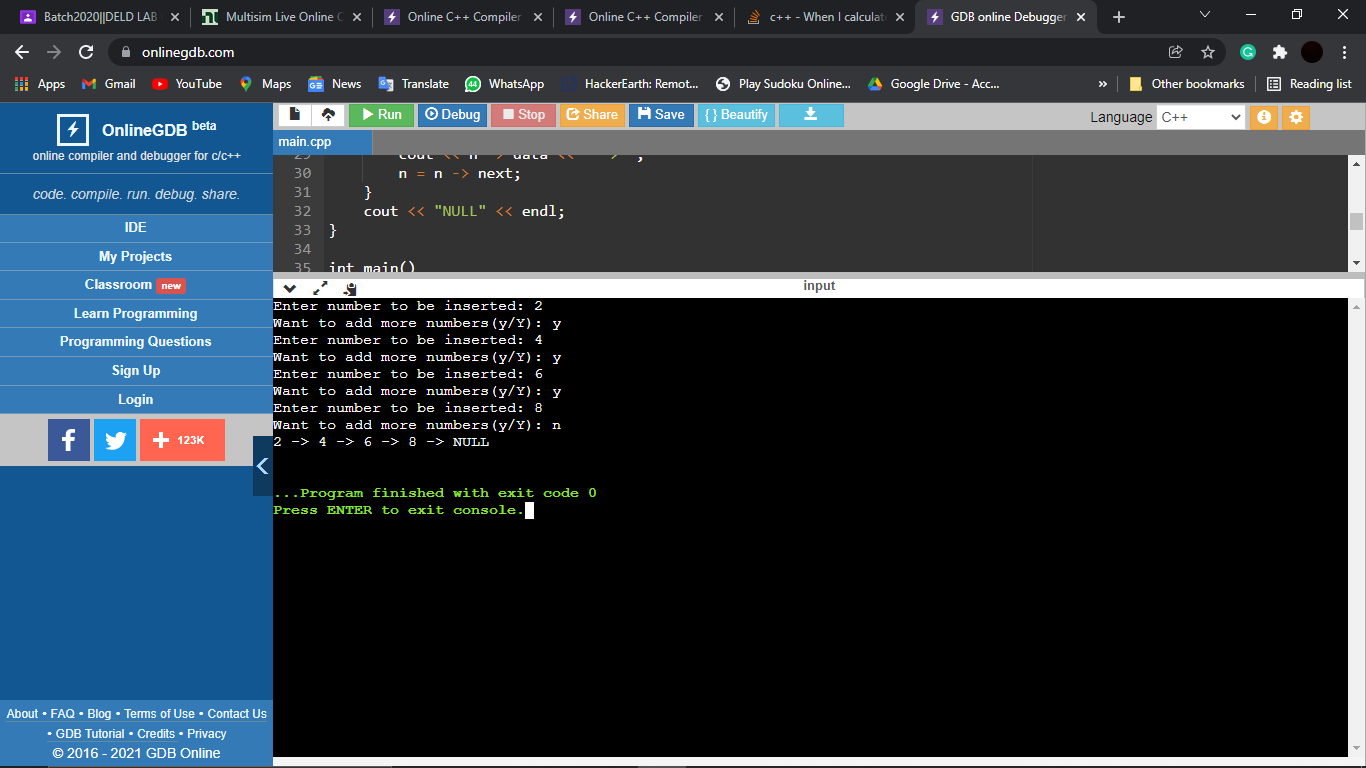
}

display(head);

return 0;

}

***Output-***



***Question 21***

Write a program to implement the following of Doubly Linked List?

***Solution-***

(A). Insert an element at beginning

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

Node\* prev;

};

void FowPrintList(Node\* n){

while(n!=NULL){

cout << n -> data << " -> ";

n = n -> next;

}

cout << endl;

}

void insertBeg(Node\* &head, int val) {

Node\* n = new Node();

n -> data = val;

n -> next = head;

n -> prev = NULL;

if(head != NULL){

head -> prev = n;

}

head = n;

}

int main()

{

Node\* head = NULL;

int i;

char ch = 'y';

while (ch == 'y' || ch == 'Y') {

cout << "Enter number to be inserted: ";

cin >> i;

insertBeg(head,i);

cout << "Want to add more numbers(y/Y): ";

cin >> ch;

}

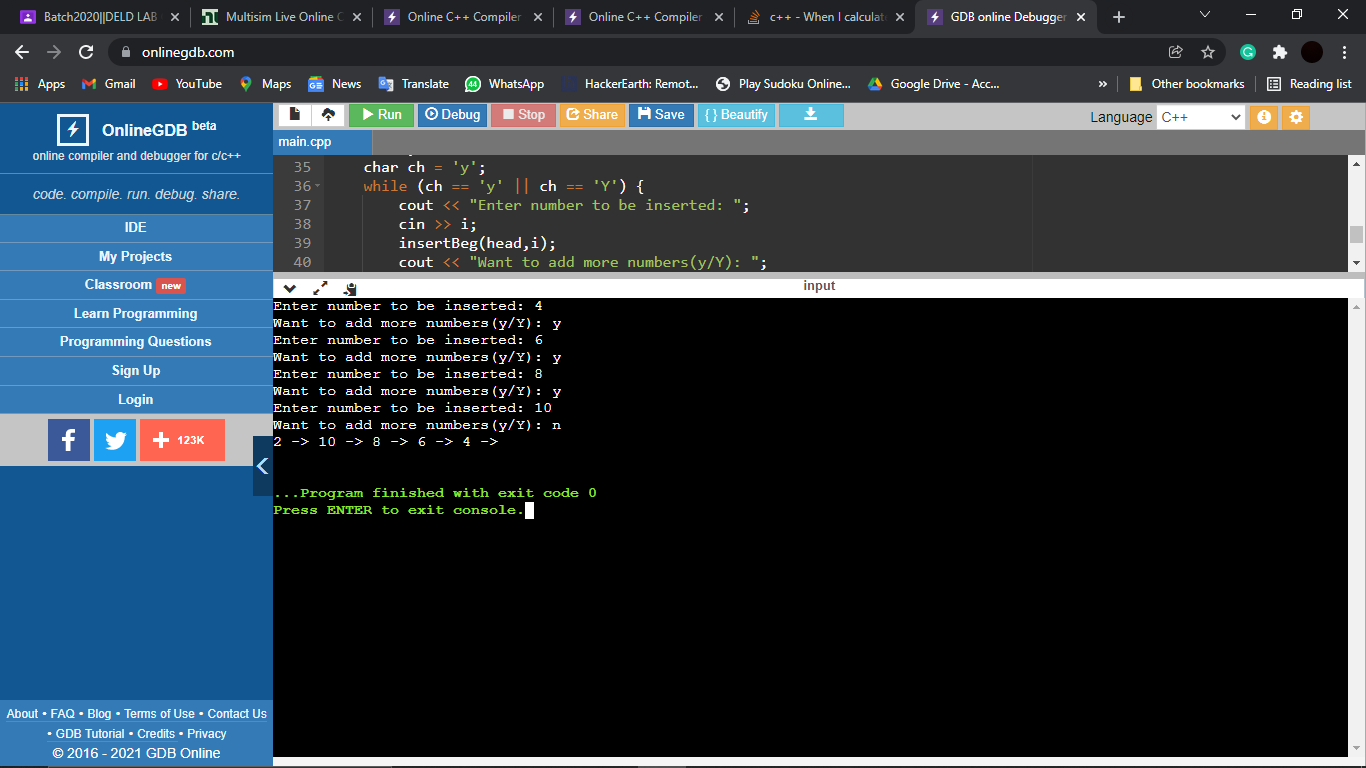
insertBeg(head,8);

FowPrintList(head);

return 0;

}

***Output-***



(B). Insert at a given Location

#include<iostream>

using namespace std;

struct Node{

int data;

Node\* prev;

Node\* next;

};

void insertBeg(Node\*\* head,int data){

Node\*node=new Node();

node->data =data;

node->next=\*head;

node->prev=NULL;

if(\*head=NULL)

(\*head)->prev=node;

\*head=node;

}

void insertAfter(Node\* prev,int data){

if(prev == NULL){

cout << "The given previous node cannot be NULL";

return;

}

Node\* node = new Node();

node -> data = data; // insert data

node -> next = prev -> next; // link new node after prev node

node -> prev = prev;

prev -> next -> prev = node;

prev -> next = node; // link the previous node to new node

}

void insertBefore(Node\* next,int data){

if (next == NULL){

cout << "The given next node cannot be NULL";

return;

}

Node\* node =new Node();

node->data = data; // Insert data in new node

next->prev->next = node; // link next of previous node to new node

node->prev = next->prev; // Link the prev of new node

next->prev = node; // Link the prev of next node

node->next = next; // Link the nextof new node to next.

}

void printList(Node\* p){

while(p!=NULL){

cout<<p->data<<" ";

p=p->next;

}

}

int main(){

Node\*head=NULL;

printList(head);

insertBeg(&head,6);

insertBeg(&head,5);

insertBeg(&head,3);

insertAfter(head->next,10); // after 2nd Node;

insertBefore(head->next -> next,13); //before 2nd Node;

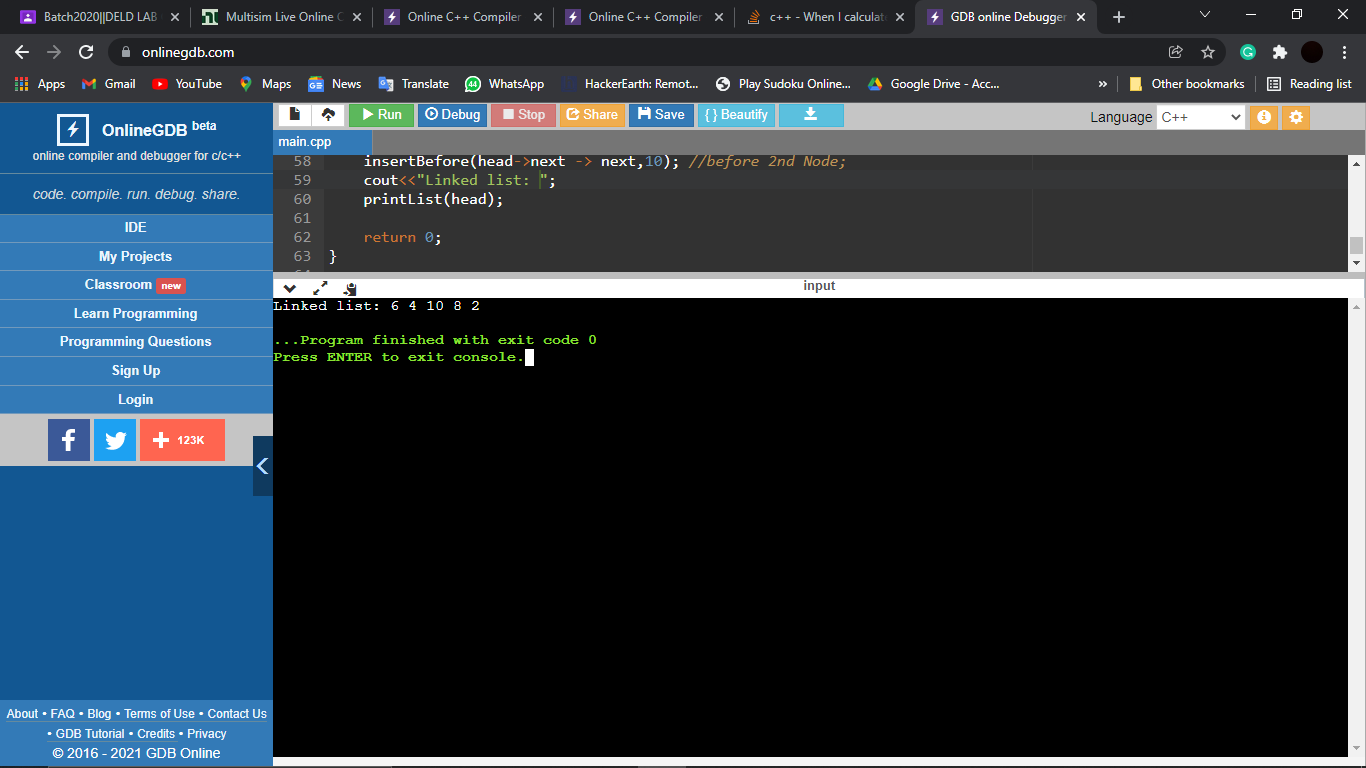
cout<<"linked list=";

printList(head);

return 0;

}

***Output-***



(C). Insert an element at the end of Doubly linked list.

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

Node\* prev;

};

void FowPrintList(Node\* n){

while(n!=NULL){

cout << n -> data << " -> ";

n = n -> next;

}

cout << endl;

}

void insertEnd(Node\* &head, int val) {

Node\* n = new Node();

Node\* ptr = head;

n -> data = val;

n -> next = NULL;

if(head == NULL) {

head = n;

n -> prev = NULL;

return;

}

while(ptr -> next != NULL){ // last node tkk traverse karte raho

ptr = ptr -> next;

}

ptr -> next = n;

n -> prev = ptr;

return;

}

int main()

{

Node\* head = NULL;

int i;

char ch = 'y';

while (ch == 'y' || ch == 'Y') {

cout << "Enter number to be inserted: ";

cin >> i;

insertEnd(head,i);

cout << "Want to add more numbers(y/Y): ";

cin >> ch;

}

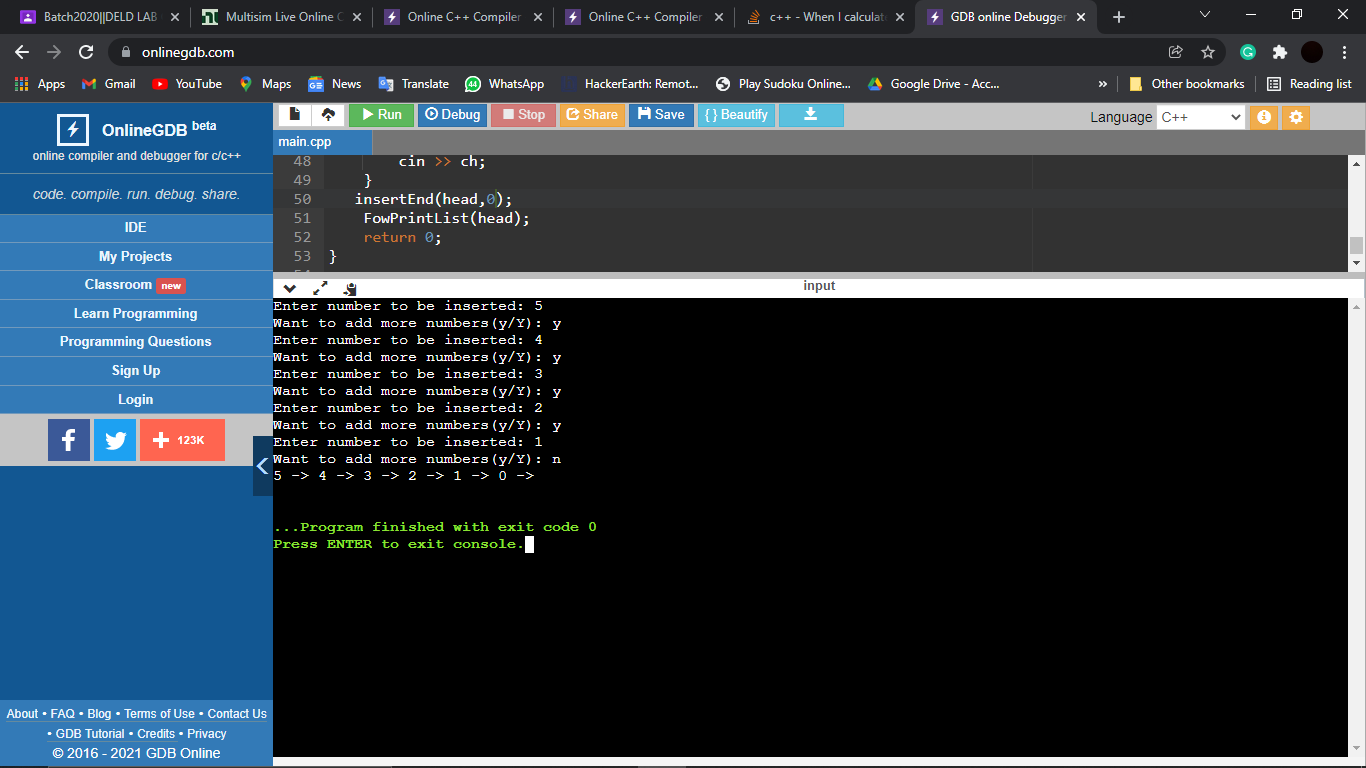
insertEnd(head,25);

FowPrintList(head);

return 0;

}

***Output-***



***Question 22***

Write a program to implement the circular Linked list?

***Solution-***

#include <iostream>

using namespace std;

class node{

public:

int data;

node\* next;

node(int val){

data = val;

next = NULL;

}

};

void insertAtHead(node\* &head, int val){

node\* n = new node(val);

if(head==NULL){

n->next = n;

head = n;

return;

}

node\* temp = head;

while(temp->next!=head){

temp=temp->next;

}

temp->next = n;

n->next = head;

head = n;

}

void insertAtTail(node\* &head, int val){

if(head==NULL){

insertAtHead(head, val);

return;

}

node\* n = new node(val);

node\* temp = head;

while(temp->next != head){

temp = temp->next;

}

temp->next = n;

n->next = head;

}

void insertAtIndex(node\* head, int data, int index){

node\* p = head;

int i = 0;

while(i!=index-1){

p = p->next;

i++;

}

node\* ptr = new node(data);

ptr->next = p->next;

p->next = ptr;

}

void deleteAtHead(node\* &head){

node\* temp = head;

while(temp->next!=head){

temp = temp->next;

}

node\* todelete = head;

temp->next = head->next;

head=head->next;

delete todelete;

}

void deletion(node\* &head, int index){

if(index==1){

deleteAtHead(head);

return;

}

node\* temp = head;

int i = 1;

while(i!=index-1){

temp = temp->next; //jo node delete krni uss se ek node pehle ruk jayega

i++;

}

node\* todelete = temp->next;

temp->next = temp->next->next;

delete todelete;

}

void display(node\* head){

node\* temp = head;

do{

cout<<temp->data<<" ";

temp=temp->next;

}

while(temp!=head);

cout<<endl;

}

int main()

{

node\* head = NULL;

insertAtTail(head, 1);

insertAtTail(head, 2);

insertAtTail(head, 13);

insertAtTail(head, 10);

display(head);

insertAtHead(head, 4);

insertAtIndex(head, 17, 1);

display(head);

deletion(head, 3); //from end

display(head);

deletion(head, 5); //from head

display(head);

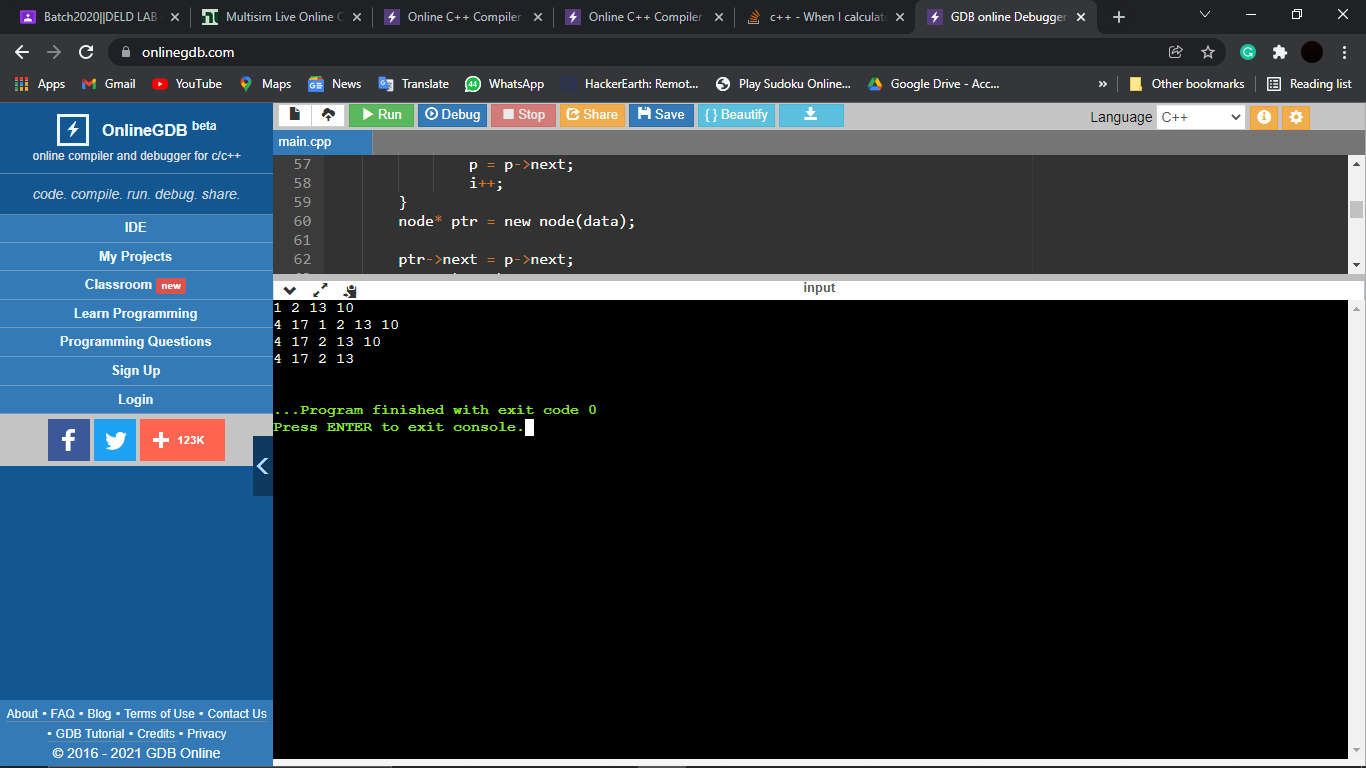
deletion(head, 9); //in between

display(head);

return 0;

}

***Output-***



***Question 23***

Write a program to implement the Pop and push operation on stack using array?

***Solution-***

#include<iostream>

using namespace std;

class Stack{

int \*stack;

int top;

int size;

int MAX\_CAPACITY;

public:

Stack(int capacity){

MAX\_CAPACITY=capacity;

stack=new int[MAX\_CAPACITY];

size=0;

top=-1;

}

void push(int data){

if(top==MAX\_CAPACITY){

cout<<"overflow";

}

else{

cout<<data<<" pushed to stack "<<endl;

stack[++top]=data;

cout<<"top is now at "<<top<<endl;

size++;

}

}

int pop(){

if(top==-1){

cout<<"underflow";

return -1;

}

else{

int temp=stack[--top];

cout<<"popped from stack "<<endl;

cout<<"top is now at "<<top<<endl;

size--;

return temp;

}

}

};

int main(){

Stack \*theStack=new Stack(100);

int temp;

theStack->push(1);

theStack->push(3);

theStack->push(6);

theStack->push(9);

// theStack->push(2);

temp=theStack->pop();

temp=theStack->pop();

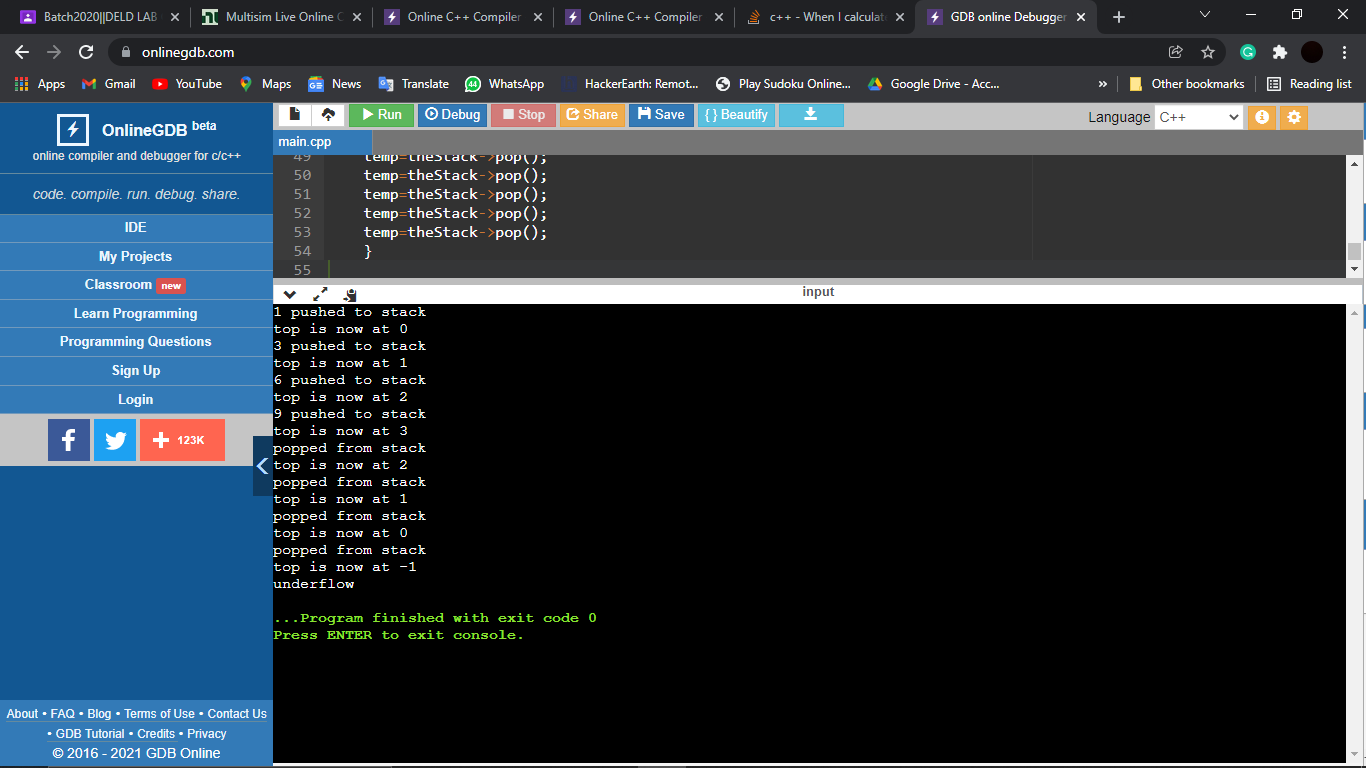
temp=theStack->pop();

temp=theStack->pop();

temp=theStack->pop();

}

***Output-***



***Question 24***

Write a program to implement push and pop operation on stack using LL?

***Solution-***

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

class Stack {

Node\* top = NULL;

public:

void push(int val) {

Node\* n = new Node();

n -> data = val;

n -> next = top;

top = n;

}

void pop() {

if (top == NULL) {

cout << "Stack is empty" << endl;

return;

}

Node\* temp = top;

top = top -> next;

delete temp;

}

int Top() {

if (top == NULL) {

cout << "Stack is empty" << endl;

return -1;

}

return top -> data;

}

bool isEmpty() {

if (top == NULL)

return true;

return false;

}

};

int main()

{

Stack s;

s.push(1);

s.push(3);

s.push(7);

s.push(9);

// Stack :- 9 -> 7 -> 3 -> 1

cout << s.Top() << endl;

s.pop(); // 9 is popped

s.pop(); // 7 is popped

cout << s.Top() << endl;

s.pop(); // 3 is popped

s.pop(); // 1 is popped

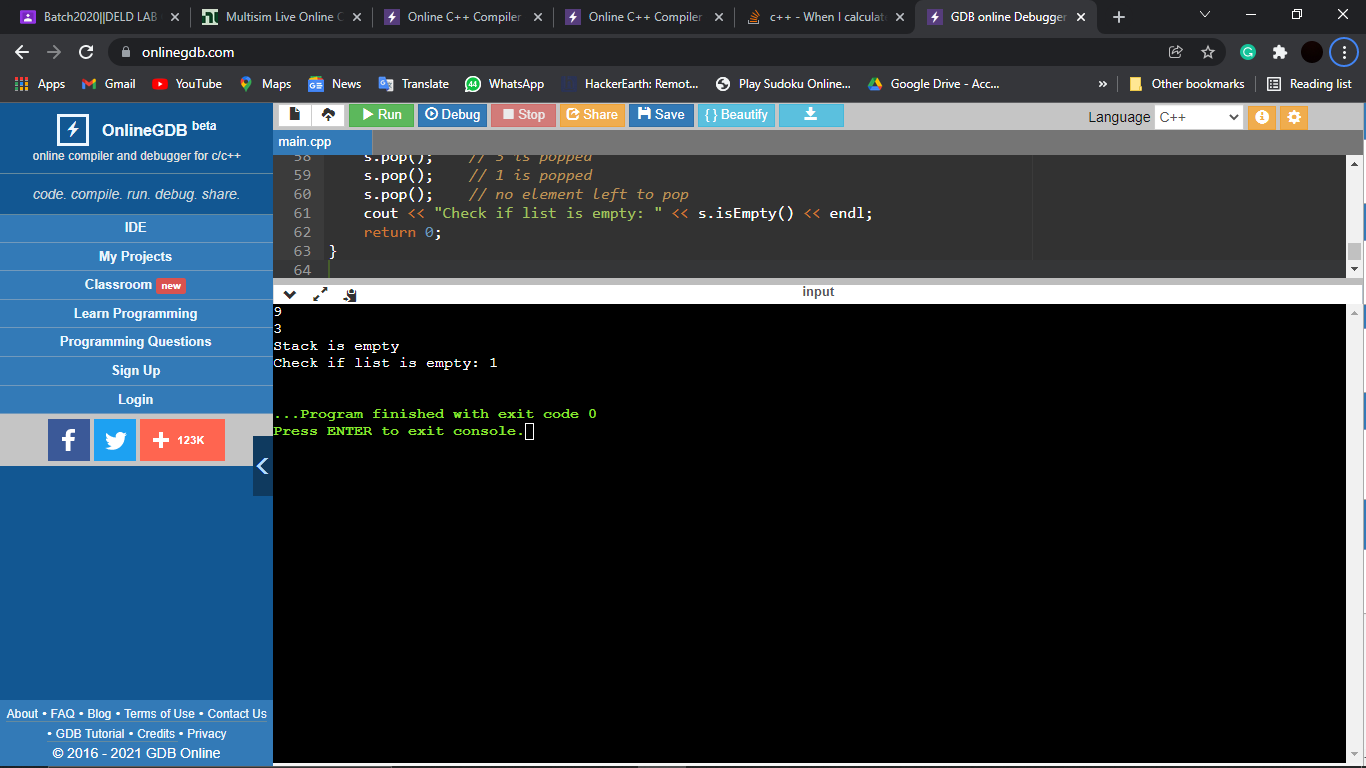
s.pop(); // no element left to pop

cout << "Check if list is empty: " << s.isEmpty() << endl;

return 0;

}

***Output-***



***Question 25***

Write a program to infix to postfix expression?

***Solution-***

#include<iostream>

#include<cstdio>

#include<cmath>

#include<stack>

using namespace std;

int isOperand(char ch) {

return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') || (isdigit(ch));

}

int checkPrecedence(char ch) {

switch (ch) {

case '+':

case '-':

return 1;

case '\*':

case '/':

return 2;

case '^':

return 3;

}

return -1;

}

int infixToPostfix(char exp[], char output[]) {

int i = 0, k = 0;

stack < char > st;

while (exp[i]) {

if (isOperand(exp[i]))

output[k++] = exp[i];

else if (exp[i] == '(')

st.push(exp[i]);

else if (exp[i] == ')') {

while (!st.empty() && st.top() != '(') {

output[k++] = st.top();

st.pop();

}

if (!st.empty() && st.top() != '(')

return -1;

else

st.pop();

} else {

while (!st.empty() && checkPrecedence(exp[i]) <= checkPrecedence(st.top())) {

output[k++] = st.top();

st.pop();

}

st.push(exp[i]);

}

i++;

}

while (!st.empty()) {

output[k++] = st.top();

st.pop();

}

output[k++] = '\0';

}

int main() {

char exp[] = "9\*(5^4)+4";

char output[20];

infixToPostfix(exp, output);

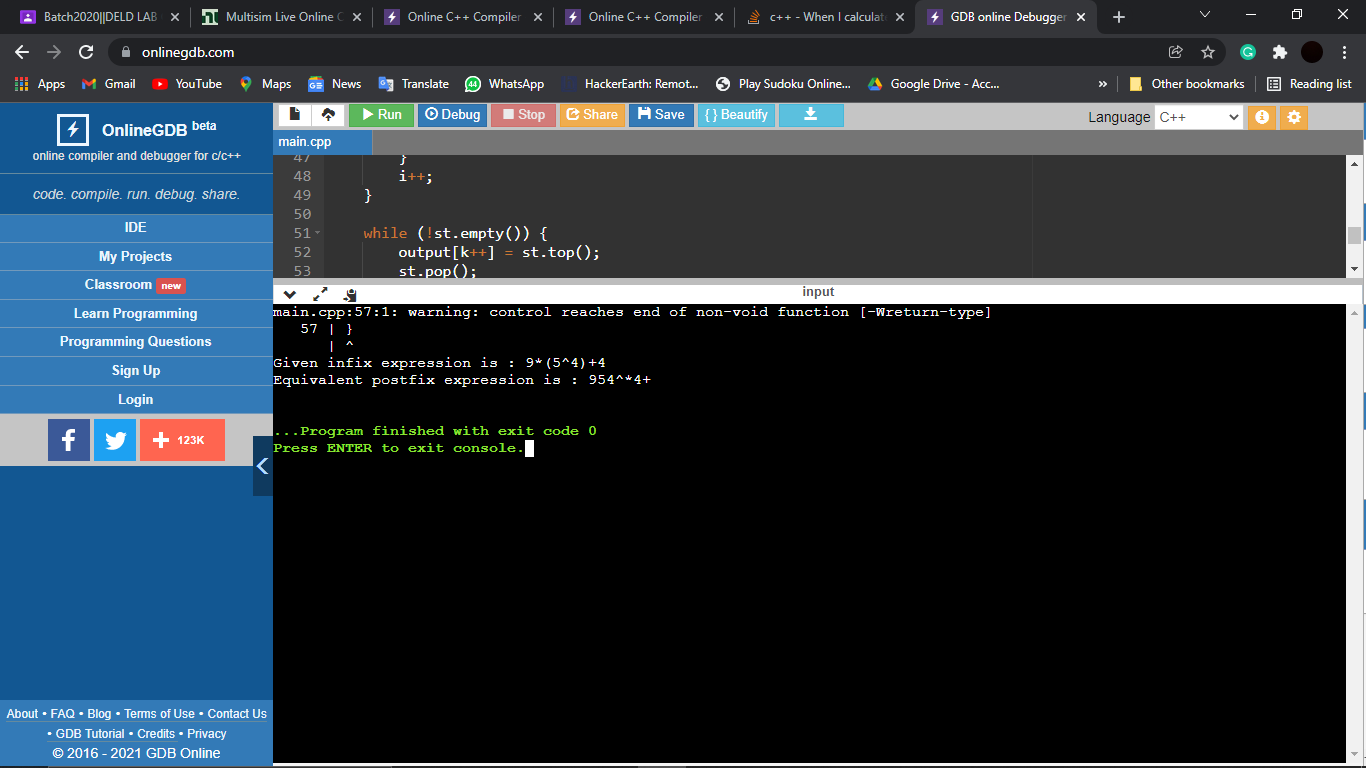
cout << "Given infix expression is : " << exp << endl;

cout << "Equivalent postfix expression is : " << output << endl;

return 0;

}

***Output-***



***Question 26***

Write a program to implement the enqueue and dequeue operation?

***Solution-***

#include <bits/stdc++.h>

#include <iostream>

using namespace std;

#define size 50

class Queue{

public:

int arr[size];

int front, rear;

Queue(){

front = -1;

rear = -1;

}

int isFull(){

if(rear == size-1){

return 1;

}

return 0;

}

int isEmpty(){

if(front == -1){

return 1;

}

return 0;

}

int enQueue(int val){

if(isFull()){

cout<<"Queue Overflow"<<endl;

return 0;

}

if(front = -1){

front = 0;

}

rear++;

arr[rear] = val;

cout<<"Element Queued at position "<<rear<<" is: "<<val<<endl;

return 0;

}

int deQueue(void){

if (isEmpty()){

cout<<"Queue Underflow"<<endl;

return 0;

}

if(front>=rear){

front = -1;

rear = -1;

cout<<"All Elements deQueued"<<endl;

return 0;

}

int n = arr[front];

cout<<"Element deQueued form position "<<front<<" is: "<<n<<endl;

front++;

// delete n;

}

void display(){

if(isEmpty()){

cout<<"Queue is Empty"<<endl;

return;

}

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

cout<<arr[i]<<endl;

}

}

};

int main()

{

Queue q;

q.deQueue();

q.enQueue(3);

q.enQueue(2);

q.deQueue();

q.deQueue();

q.enQueue(4);

q.enQueue(6);

q.enQueue(0);

q.enQueue(8);

q.enQueue(3);

q.enQueue(1);

q.display();

q.deQueue();

q.deQueue();

q.deQueue();

q.deQueue();

q.deQueue();

q.deQueue();

q.deQueue();

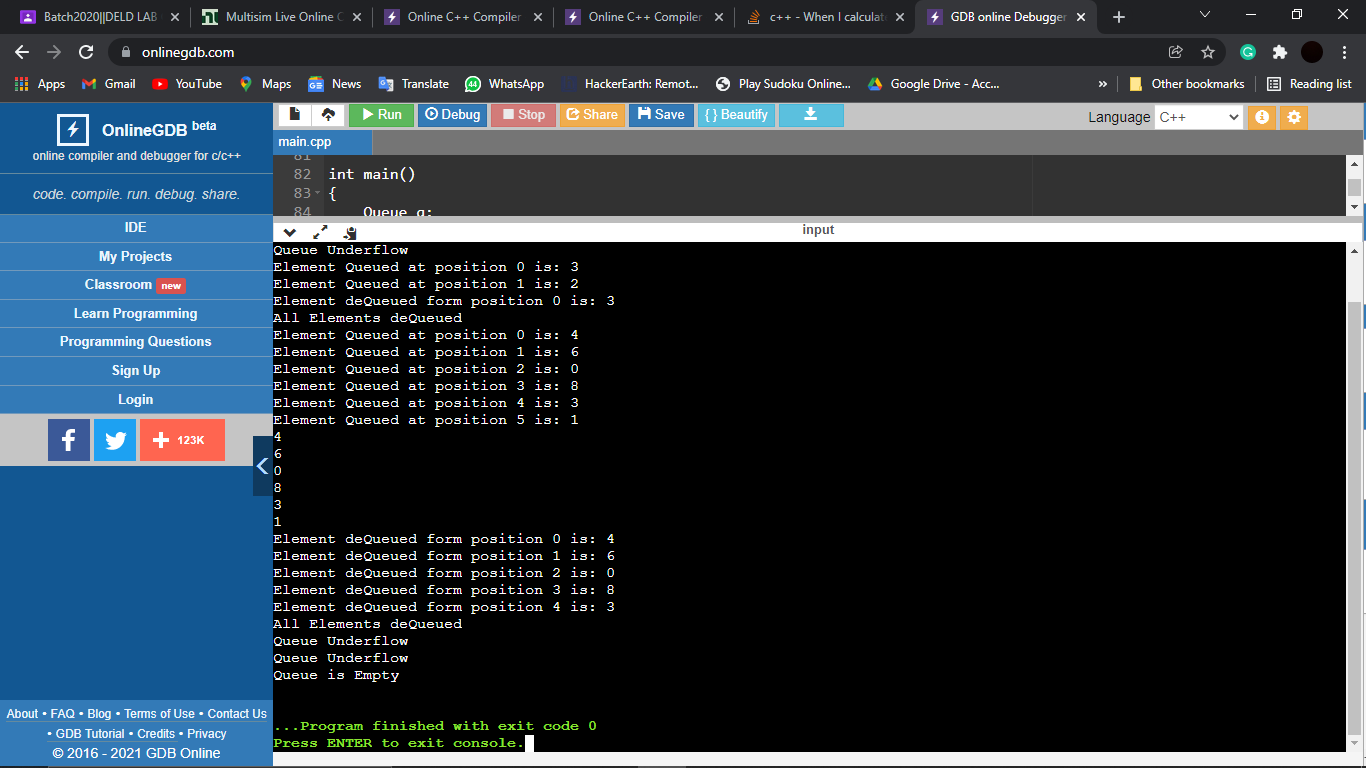
q.deQueue();

q.display();

return 0;

}

***Output-***



***Question 27***

Write a Program to implement the reverse of given exp. Or string using stack?

***Solution-***

#include <bits/stdc++.h>

using namespace std;

class Stack

{

public:

int top;

unsigned capacity;

char\* array;

};

Stack\* createStack(unsigned capacity)

{

Stack\* stack = new Stack();

stack->capacity = capacity;

stack->top = -1;

stack->array = new char[(stack->capacity \* sizeof(char))];

return stack;

}

int isFull(Stack\* stack)

{ return stack->top == stack->capacity - 1; }

int isEmpty(Stack\* stack)

{ return stack->top == -1; }

void push(Stack\* stack, char item)

{

if (isFull(stack))

return;

stack->array[++stack->top] = item;

}

char pop(Stack\* stack)

{

if (isEmpty(stack))

return -1;

return stack->array[stack->top--];

}

void reverse(char str[])

{

int n = strlen(str);

Stack\* stack = createStack(n);

int i;

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

push(stack, str[i]);

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

str[i] = pop(stack);

}

int main()

{

char str[] = "ChitkaraUniversity";

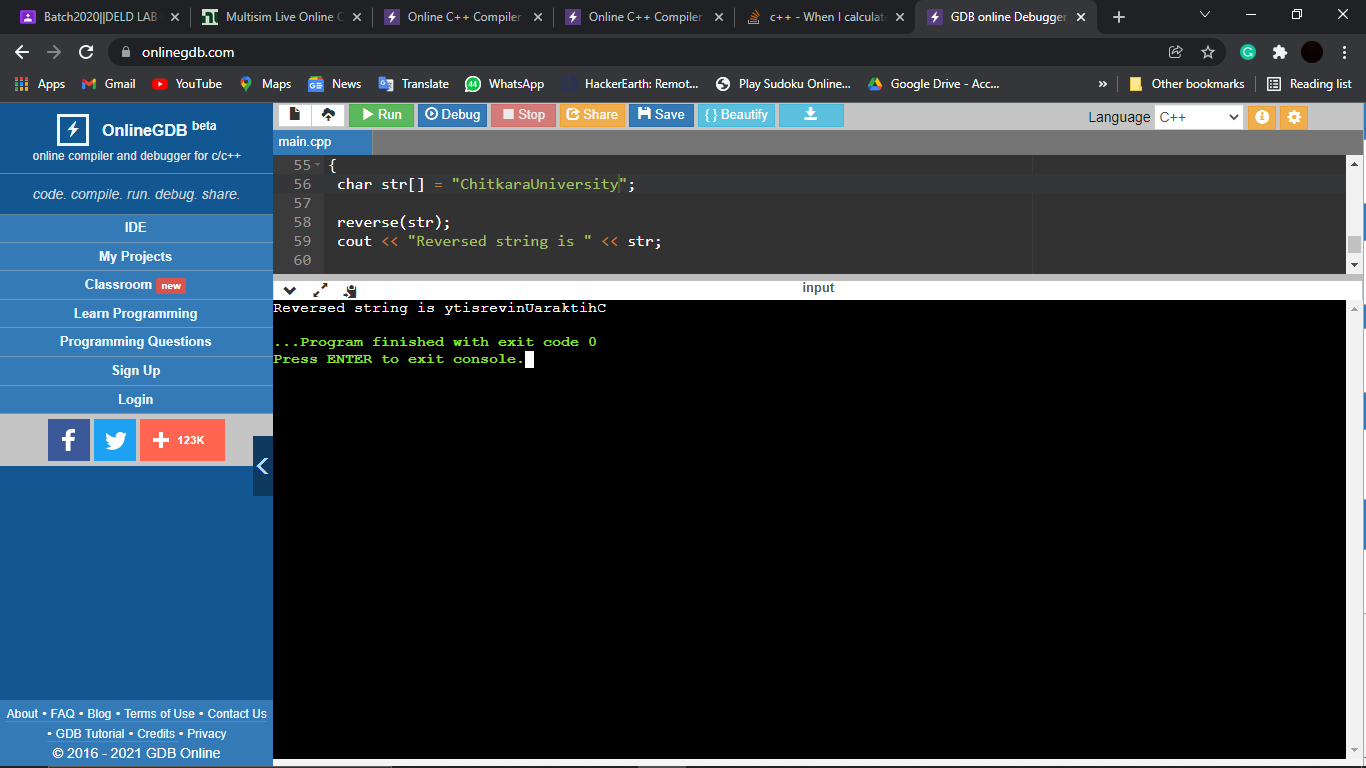
reverse(str);

cout << "Reversed string is " << str;

return 0;

}

***Output-***



***Question 28***

Write a program to implement the tower of Hanoi Problem (No. of disc entered by user, using recursion)?

***Solution-***

#include <iostream>

using namespace std;

void tower(int n, char source, char des, char via){

if(n == 1){

cout << "Moved the ring "<< n << " from " << source << " to " << des << " via " << via << ".\n";

}

else{

tower(n-1,source, via, des);

cout << "Moved the ring " << n << " from " << source << " to " << des << " via " << via << ".\n";

tower(n-1,via,des,source);

}

}

int main()

{

int rings;

char source='A';

char des = 'B';

char via = 'C';

cout << "Enter the number of rings : ";

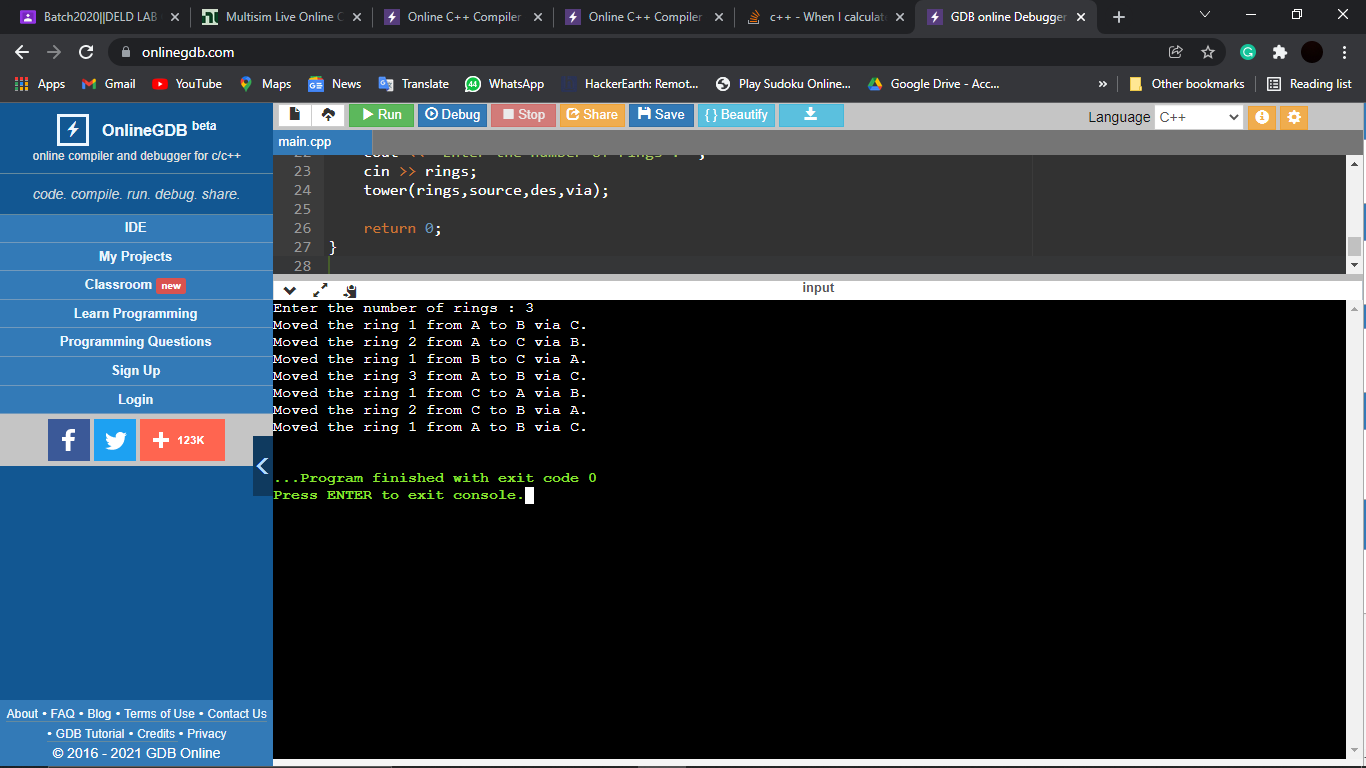
cin >> rings;

tower(rings,source,des,via);

return 0;

}

***Output-***



***Question 29***

Write a program to implement priority queue?

***Solution-***

#include <bits/stdc++.h>

using namespace std;

class item {

public:

int value;

int priority;

};

item pr[100000];

int size = -1;

void enqueue(int value, int priority)

{

size++;

pr[size].value = value;

pr[size].priority = priority;

}

int peek()

{

int highestPriority = INT\_MIN;

int ind = -1;

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

if (highestPriority

== pr[i].priority

&& ind > -1

&& pr[ind].value

< pr[i].value) {

highestPriority = pr[i].priority;

ind = i;

}

else if (highestPriority

< pr[i].priority) {

highestPriority = pr[i].priority;

ind = i;

}

}

return ind;

}

void dequeue()

{

int ind = peek();

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

pr[i] = pr[i + 1];

}

size--;

}

int main()

{

enqueue(10, 2);

enqueue(14, 4);

enqueue(16, 4);

enqueue(12, 3);

int ind = peek();

cout << pr[ind].value << endl;

dequeue();

ind = peek();

cout << pr[ind].value << endl;

dequeue();

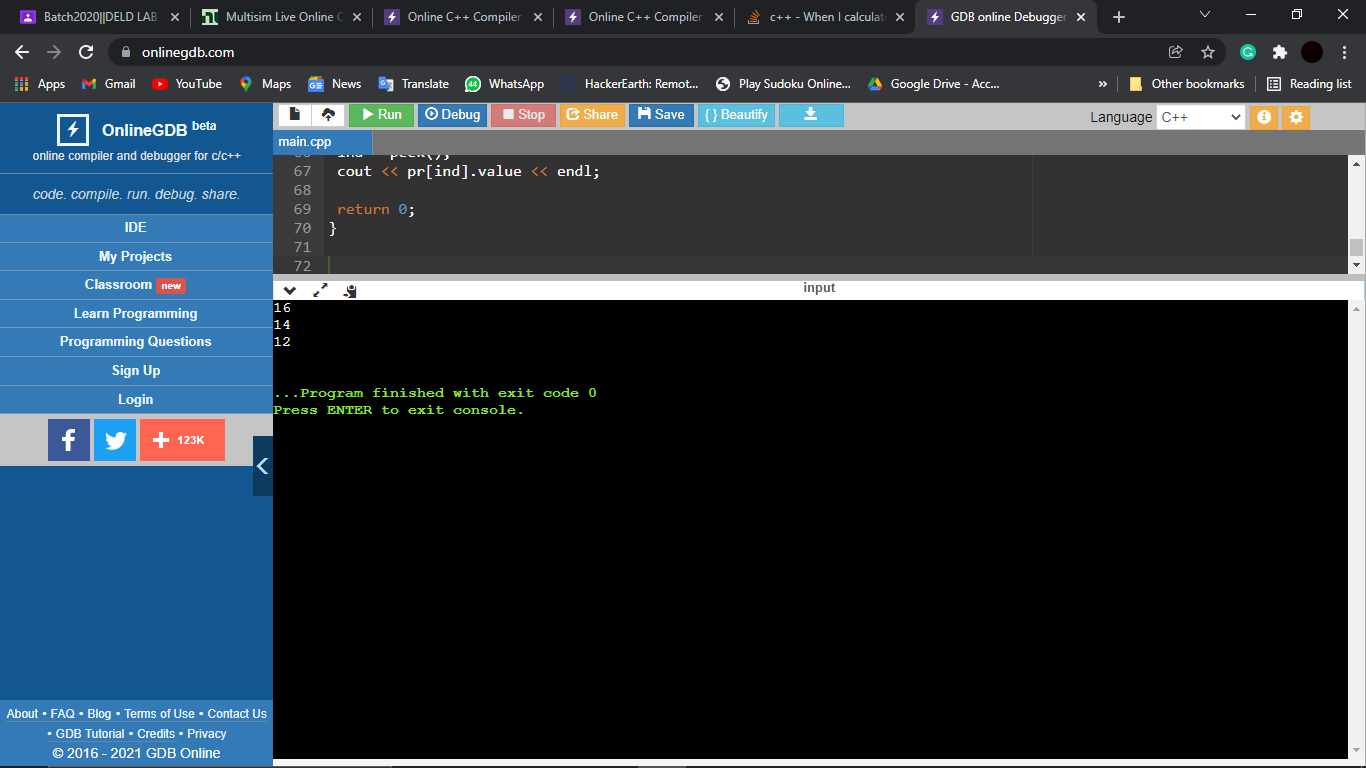
ind = peek();

cout << pr[ind].value << endl;

return 0;

}

***Output-***



***Question 30***

Write a program to implement Bubble Sort?

***Solution-***

#include<iostream>

#include<cstdio>

#include<cmath>

using namespace std;

#define SIZE 10

int front=-1, rear=-1;

int circular[SIZE];

// Method to add an item to the queue.

void enqueue(int item)

{

if ((front == 0 && rear == SIZE-1) || (rear == (front-1)%(SIZE-1))) { // Queue is full

cout<<"Queue is FULL.\n";

return;

}

if(front == -1) // Insert first element

front = rear = 0;

else if (rear == SIZE-1 && front != 0) // insert the element back at starting of queue.

rear = 0;

else // insert the element normally

rear++;

circular[rear] = item;

cout<<item<<" enqueued to queue.\n";

}

// Method to remove an item from queue.

int dequeue()

{

if (front == -1)

{

cout<<"Queue is Empty.\n";

return -1;

}

int item = circular[front];

if (front == rear)

front = rear = -1;

else if (front == SIZE-1)

front = 0;

else

front++;

cout <<item<<" dequeued from queue.\n";

return item;

}

int main(){

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

enqueue(10);

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

enqueue(20);

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

enqueue(30);

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

enqueue(40);

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

dequeue();

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

dequeue();

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

dequeue();

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

dequeue();

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

dequeue();

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

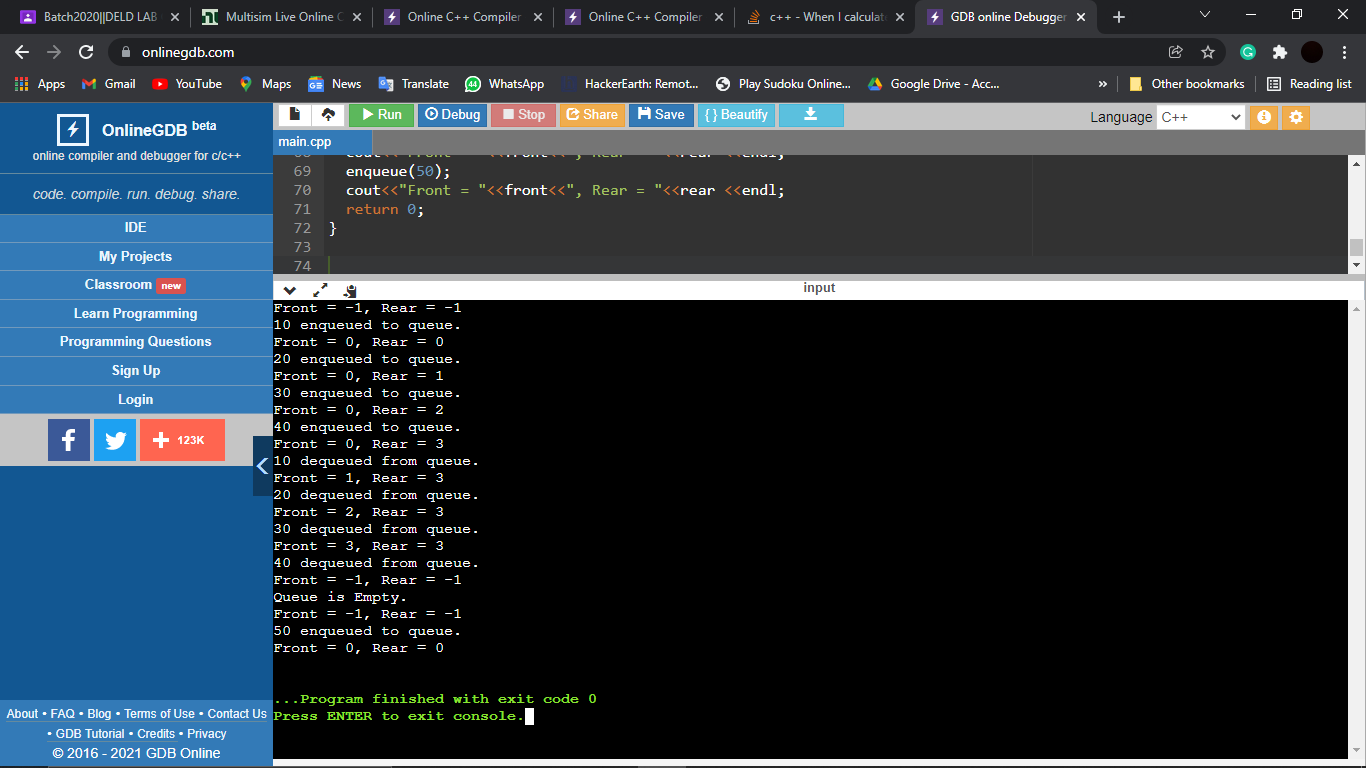
enqueue(50);

cout<<"Front = "<<front<<", Rear = "<<rear <<endl;

return 0;

}

***Output-***



***Question 31***

Write a program to implement Bubble Sort?

***Solution-***

#include <iostream>

using namespace std;

class array{

public:

int \*arr;

int size;

array(int size){

arr = new int [size];

}

void input(){

cout << "Enter the elements of array : " << endl;

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

cin >> arr[i];

}

}

void bubblesort(){

for(int i = 1;i<size;i++){

for(int j = 1;j<=size-i;j++){

if(arr[j-1]>arr[j]){

int temp = arr[j];

arr[j] = arr[j-1];

arr[j-1] = temp;

}

}

cout << "Array after " << i <<" Pass" <<endl;

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

cout << arr[i] << " ";

}

cout << endl;

}

}

void display(){

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

cout << arr[i] << " ";

}

cout << endl;

}

};

int main(){

int n;

cout << "Enter the size of array : ";

cin >> n;

array a1(n);

a1.size = n;

a1.input();

cout << "Array before sorting is : " << endl;

a1.display();

cout << endl;

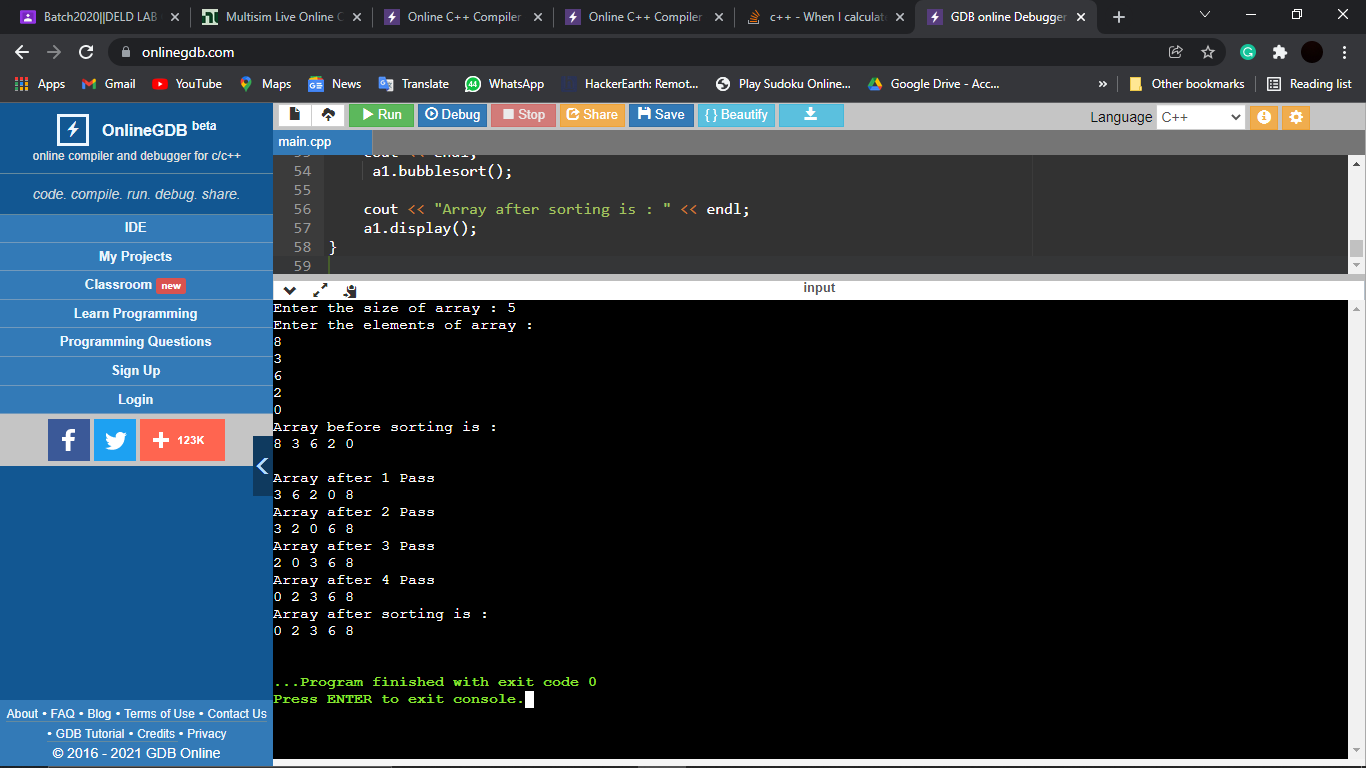
a1.bubblesort();

cout << "Array after sorting is : " << endl;

a1.display();

}

***Output-***



***Question 32***

Write a program to implement the insertion, selection, merge, quick sort.

***Solution:***

***Insertion Sort***

#include <iostream>

using namespace std;

int main()

{

int p[100],i,j,k,n;

cout<<"Enter the size of array: ";

cin>>n;

cout<<"Enter the elements: ";

for(i=0;i<=n-1;i++){

cin>>p[i];

}

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

for(j=i;j>0;j--){

if(p[j]<p[j-1]){

k=p[j];

p[j]=p[j-1];

p[j-1]=k;

}

}

}

cout<<"Array after Insertion sort: ";

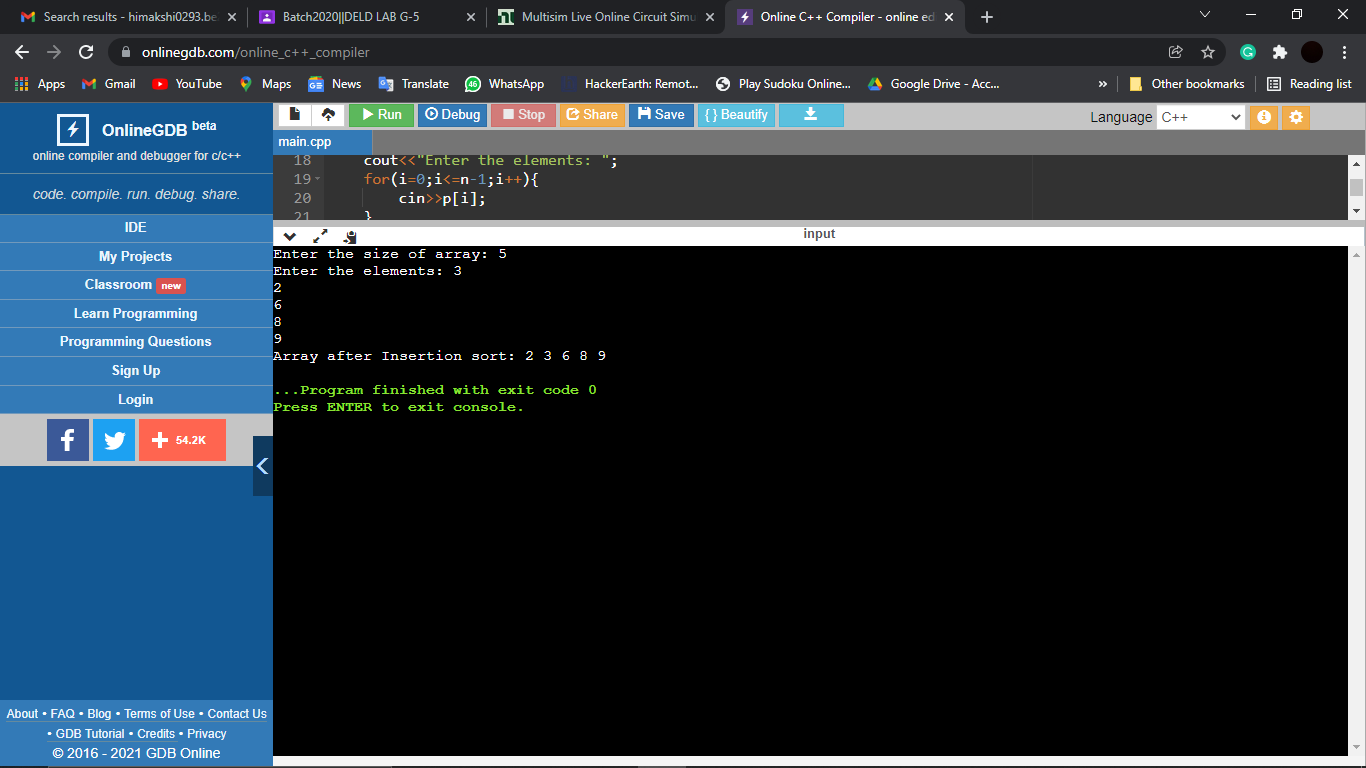
for(i=0;i<=n-1;i++){

cout<<p[i]<<" ";

}

return 0;

}



***Selection Sort***

#include <iostream>

using namespace std;

class array{

public:

int \*arr;

int size;

array(int size){

arr = new int [size];

}

void input(){

cout << "Enter the elements of array : " << endl;

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

cin >> arr[i];

}

}

void selectionSort(){

for(int i = 0;i<size-1;i++){

int min = i;

for(int j = i+1;j<size;j++){

if(arr[j]<arr[min]){

min = j;

}

}

int temp =arr[min];

arr[min] = arr[i];

arr[i] = temp;

cout << "Array after " << i+1 << " Pass is : " << endl;

display();

}

}

void display(){

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

cout << arr[i] << " ";

}

cout << endl;

}

};

int main(){

int n;

cout << "Enter the size of array : ";

cin >> n;

array a1(n);

a1.size = n;// saare function mein utha lega value

a1.input();

cout << "Array before sorting is : " << endl;

a1.display();

cout << endl;

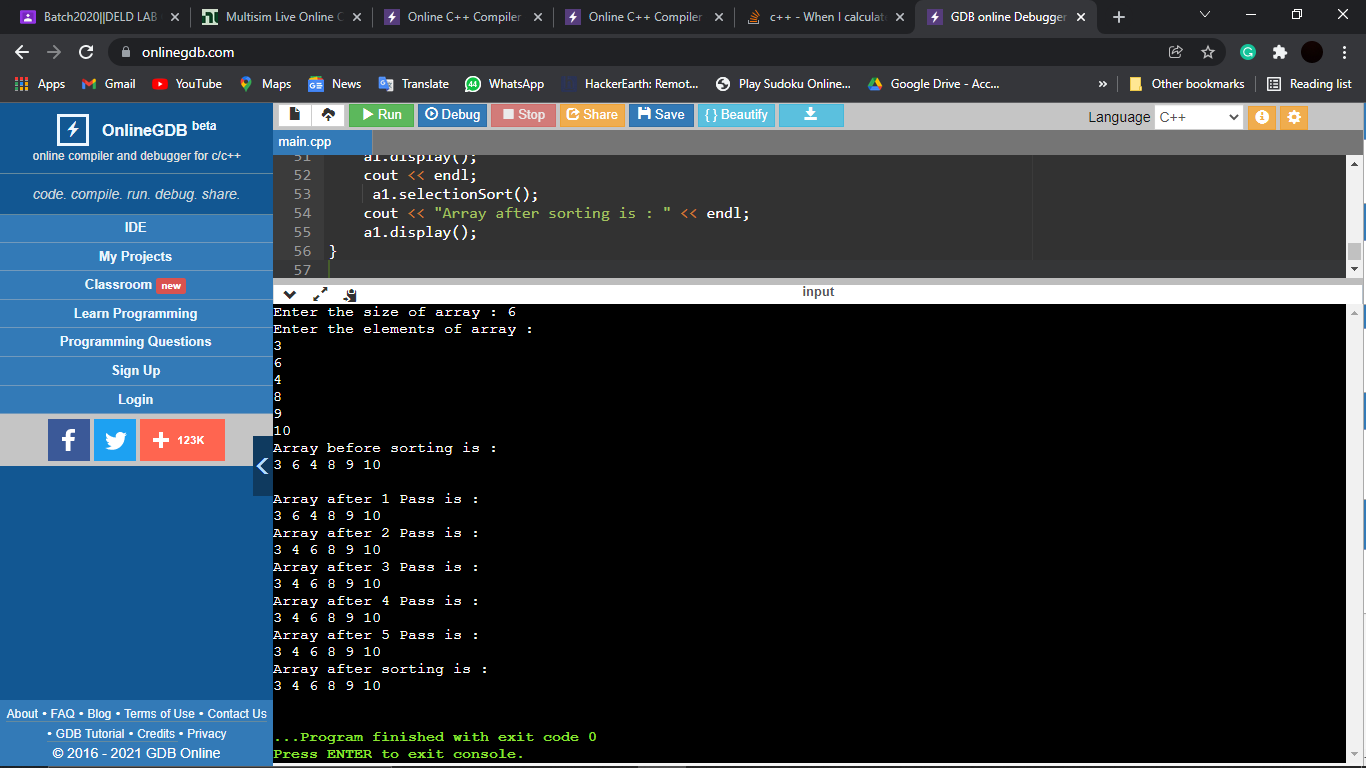
a1.selectionSort();

cout << "Array after sorting is : " << endl;

a1.display();

}

***Output-***



***Merge Sort***

#include<iostream>

using namespace std;

class MergeSorting{

public:

void Merge(int a[], int mid, int low, int high){

int b[high];

int i, j, k;

i = low;

j = mid+1;

k = low;

while(i<=mid && j<=high){

if(a[i]<a[j]){

b[k] = a[i];

k++;

i++;

}else{

b[k] = a[j];

k++;

j++;

}

}

while(i<=mid){

b[k] = a[i];

k++;

i++;

}

while(j<=high){

b[k] = a[j];

k++;

j++;

}

for(int i=low; i<= high; i++){

a[i] = b[i];

}

}

void MergeSort(int a[], int low, int high){

if(low<high){

int mid = (low+high)/2;

MergeSort(a, low, mid);

MergeSort(a, mid+1, high);

Merge(a, mid, low, high);

}

}

};

int main(){

int n;

cout<<"Enter the size of the array: ";

cin>>n;

int arr[n];

cout<<"Enter the elements of the array"<<endl;

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

cin>>arr[i];

}

MergeSorting A1;

A1.MergeSort(arr, 0, n-1);

cout << "Sorted Merge Array is : " << endl;

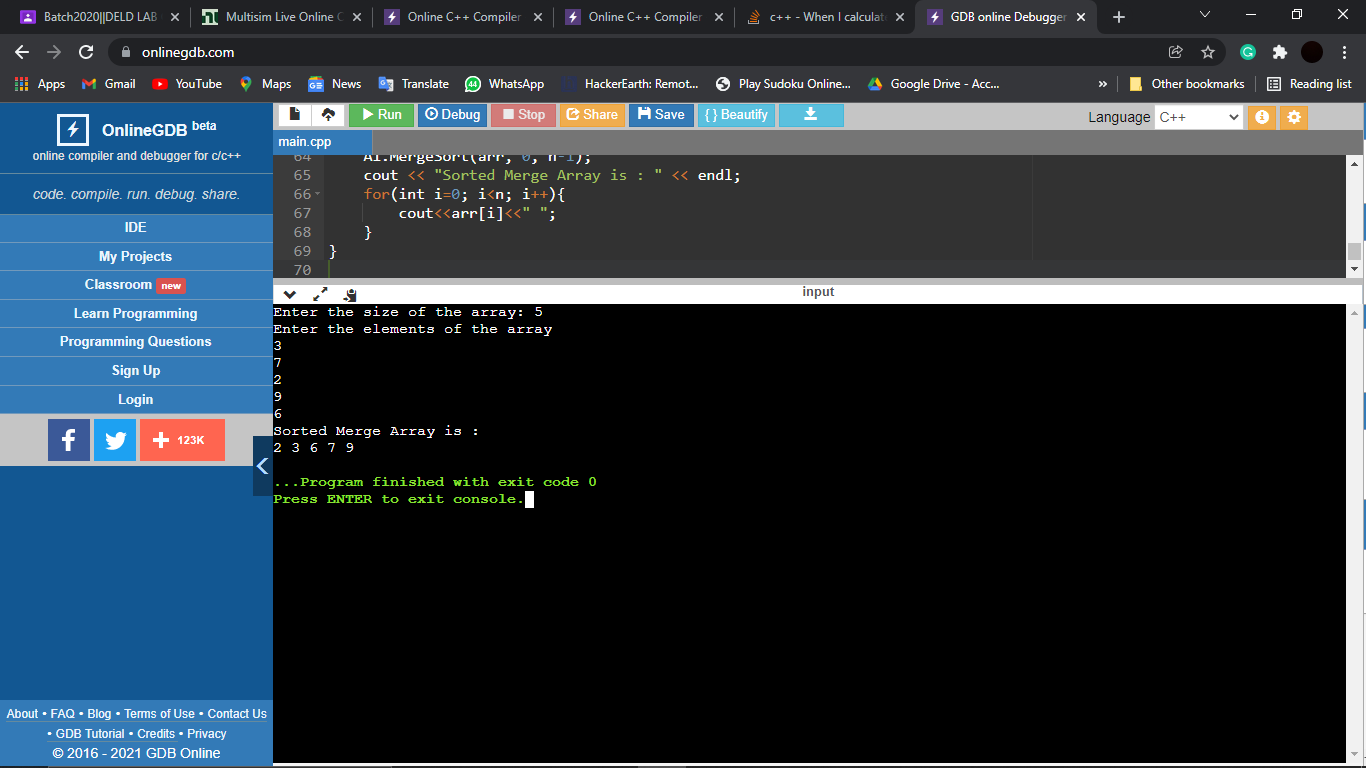
for(int i=0; i<n; i++){

cout<<arr[i]<<" ";

}

}

***Output-***



***Quick Sort***

#include <iostream>

using namespace std;

int partition(int arr[],int b,int e){

int pivot=arr[e];

int pindex=b;

for (int i = b; i < e; i++)

{

if (arr[i]<pivot)

{

int temp=arr[pindex];

arr[pindex]=arr[i];

arr[i]=temp;

pindex++;

}

}

int temp=arr[e];

arr[e]=arr[pindex];

arr[pindex]=temp;

return pindex;

}

void quickSort(int arr[], int b, int e)

{

if (b < e)

{

int p = partition(arr, b, e);

quickSort(arr, b, p-1);

quickSort(arr, p +1, e);

}

}

int main(){

int arr[] = {0, 7, 8, 9, 1, 5};

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

cout<<n;

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

{

cout<<arr[i]<<endl;

}

quickSort(arr, 0, n );

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

{

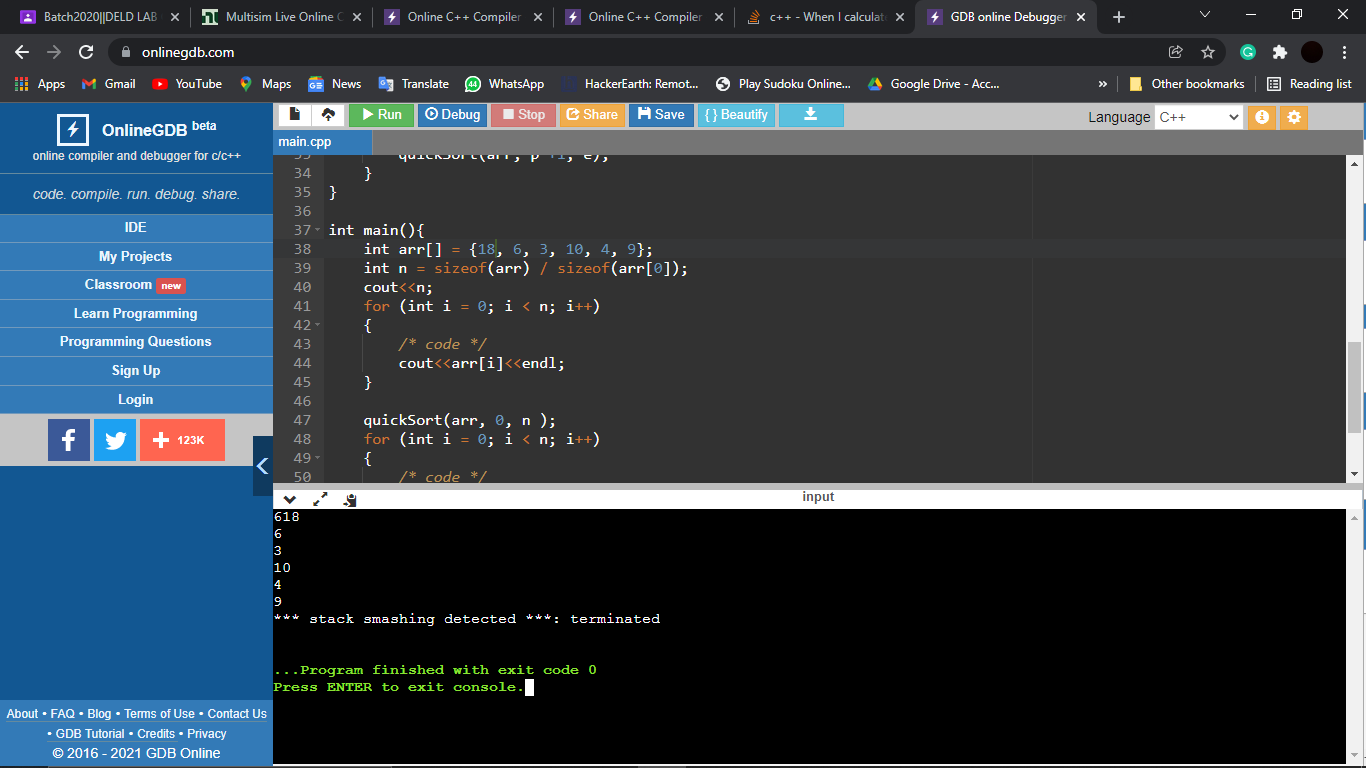
cout<<arr[i]<<" ";

}

return 0;

}

***Output-***



***Heap Sort***

#include <iostream>

using namespace std;

void MaxHeapify(int a[], int i, int n)

{

int j, temp;

temp = a[i];

j = 2\*i;

while (j <= n)

{

if (j < n && a[j+1] > a[j])

j = j+1;

// Break if parent value is already greater than child value.

if (temp > a[j])

break;

// Switching value with the parent node if temp < a[j].

else if (temp <= a[j])

{

a[j/2] = a[j];

j = 2\*j;

}

}

a[j/2] = temp;

return;

}

void HeapSort(int a[], int n)

{

int i, temp;

for (i = n; i >= 2; i--)

{

// Storing maximum value at the end.

temp = a[i];

a[i] = a[1];

a[1] = temp;

// Building max heap of remaining element.

MaxHeapify(a, 1, i - 1);

}

}

void Build\_MaxHeap(int a[], int n)

{

int i;

for(i = n/2; i >= 1; i--)

MaxHeapify(a, i, n);

}

int main()

{

int n, i;

cout<<"\nEnter the number of data element to be sorted: ";

cin>>n;

n++;

int arr[n];

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

{

cout<<"Enter element "<<i<<": ";

cin>>arr[i];

}

// Building max heap.

Build\_MaxHeap(arr, n-1);

HeapSort(arr, n-1);

// Printing the sorted data.

cout<<"\nSorted Data ";

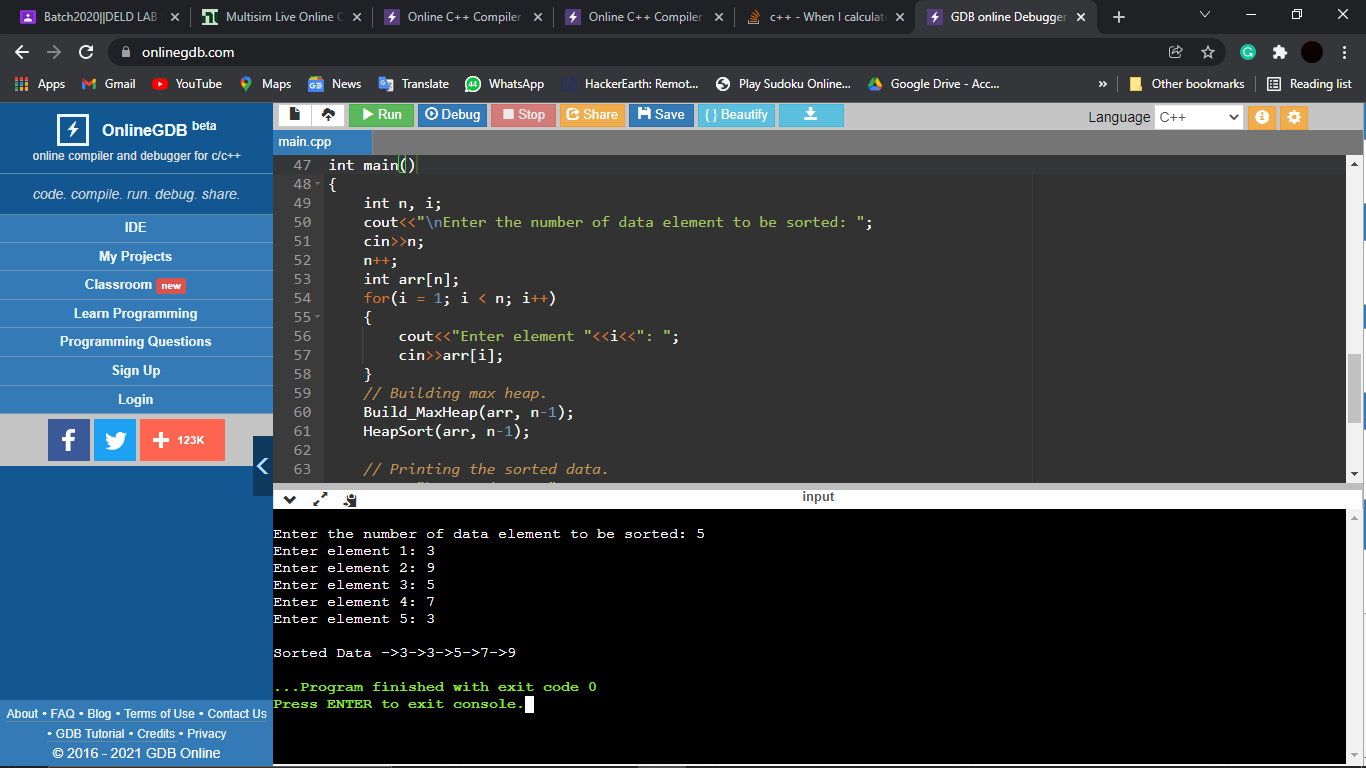
for (i = 1; i < n; i++)

cout<<"->"<<arr[i];

return 0;

}

***Output-***



***-------------X-------------X-------------X-------------Thank You-------------X-------------X-------------X-------------***