Name : Aritra Das

Roll No: 75

Assignment 3



#include <stdio.h>

#include <iostream>

using namespace std;

#define MAX(x,y) ((x>y)?x:y)

void integr(){

    int a,b,c;

    cout<<"Enter numbers to compare\n";

    cin>>a>>b;

    c= MAX(a,b);

    cout<<endl<<"The result is \n"<<c;

}

void chr(){

    char \*a,\*b;

    int c;

    cout<<"Enter numbers to compare\n";

    cin>>\*a>>\*b;

    c= MAX(\*a,\*b);

    cout<<endl<<"The result is \n"<<c;

}

int main(){

    for(;;){

        cout<<"Enter : \n1 For integer argument\n2. For character argument\n3.To exit\n";

        int c;

        cin>>c;

        switch(c){

            case 1:{

                integr();

                break;

            }

            case 2:{

                chr();

                break;

            }

            case 3:{

                cout<<"Exiting ...\n";

                exit(0);

            }

            default:{

                cout<<"Wrong input";

                break;

            }

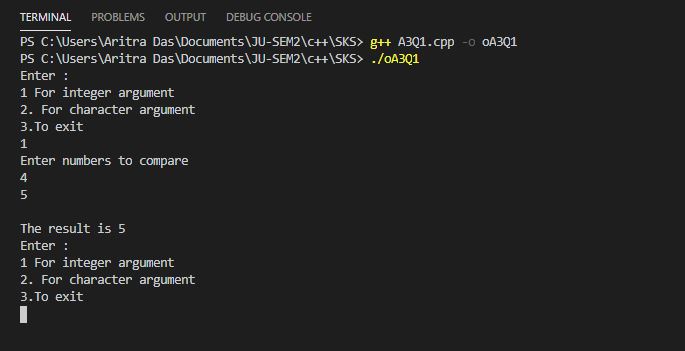
        }

    }

    return 0;

}

Output:



2.

#include <stdio.h>

#include <iostream>

using namespace std;

inline double multiply(double a, double b){

    return a\*b;

}

 double multi(double a, double b){

    return a\*b;

}

int main(){

    for(;;){

        cout<<"Enter : \n1 For normal function\n2. For inline function\n3.To exit\n";

        int c;

        cin>>c;

        switch(c){

            case 1:{

                double a,b,c;

                cout<<"Enter numbers to multiply"<<endl;

                cin>>a>>b;

                c=multi(a,b);

                cout<<"Result is  : "<<c;

                break;

            }

            case 2:{

                double a,b,c;

                cout<<"Enter numbers to multiply"<<endl;

                cin>>a>>b;

                c=multiply(a,b);

                cout<<"Result is  : "<<c;

                break;

            }

            case 3:{

                cout<<"Exiting ...\n";

                exit(0);

            }

            default:{

                cout<<"Wrong input";

                break;

            }

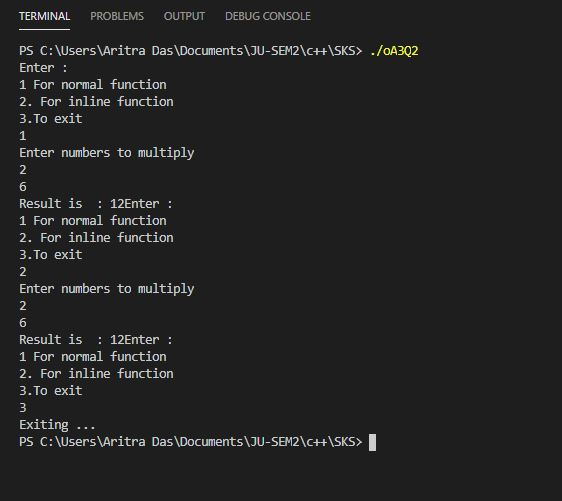
        }

    }

    return 0;

}

Output:



3.

#include <stdio.h>

#include <iostream>

//#include<bits/stdc++.h>

using namespace std;

void swap(int &a, int &b){

    a=a+b;

    b=a-b;

    a=a-b;

}

int main(){

    int a,b;

    cout<<"Enter numbers to swap"<<endl;

    cin>>a>>b;

    cout<<"Before swap : "<<a<<" "<<b<<endl;

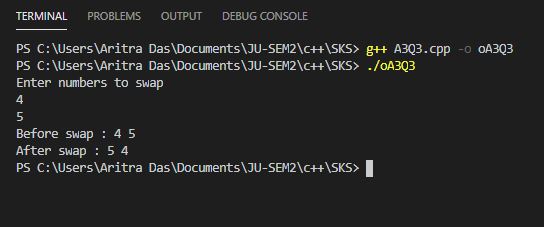
    swap(a,b);

    cout<<"After swap : "<<a<<" "<<b<<endl;

    return 0;

}

Output:



4.

#include <stdio.h>

#include <iostream>

using namespace std;

int\* max(int a,int b){

    return a>b?&a:&b;

}

int main(){

    int a,b,\*x,y;

    cout<<"Enter numbers"<<endl;

    cin>>a>>b;

    x=max(a,b);

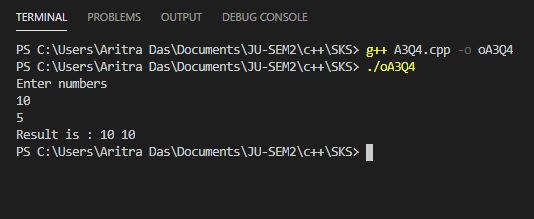
    y=\*max(a,b);

    cout<<"Result is : "<<\*x<<" "<<y;

    return 0;

}

Output:



5.

#include <iostream>

using namespace std;

double calc(double i,double t=0.1){

    return (t\*i);

}

int main(){

    double i,t;

    cout<<"Enter income and tax rate";

    cin>>i>>t;

    t/=100;

    double res= calc(i,t);

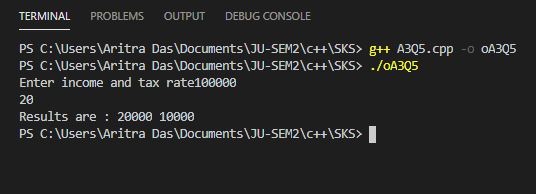
    double res2=calc(i);

    cout<<"Results are : "<<res<<" "<<res2;

    return 0;

}

Output:



6.

#include <iostream>

using namespace std;

void f(int a,int b){

    cout<<endl<<a<<" "<<b;

}

void f(char a,int b){

    cout<<endl<<a<<" "<<b;

}

int main(){

    int a;

    char b;

    float c;

    cout<<"Enter int char and float"<<endl;

    cin>>a>>b>>c;

    f(a,b);

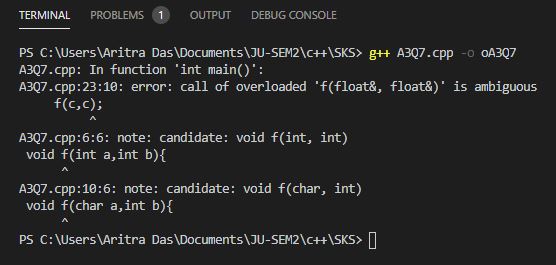
    f(b,b);

    f(c,c);

    return 0;

}

Output:



7.

#include <iostream>

using namespace std;

void f(int a,int b){

    cout<<endl<<a<<" "<<b;

}

void f(char a,int b){

    cout<<endl<<a<<" "<<b;

}

int main(){

    int a;

    char b;

    float c;

    cout<<"Enter int char and float"<<endl;

    cin>>a>>b>>c;

    f(a,b);

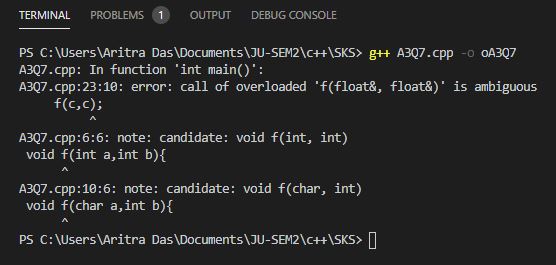
    f(b,b);

    f(c,c);

    return 0;

}

Output:



8.

#include <iostream>

using namespace std;

struct stud{

    int roll;

    double score;

    void inpt(struct stud &temp){

        cout<<"Enter roll and score"<<endl;

        cin>>temp.roll>>temp.score;

    }

    void dis(struct stud &temp){

        cout<<"Roll is : "<<temp.roll<<" and score is : "<<temp.score<<endl;

    }

};

void modify(struct stud &temp){

    cout<<"Enter new score"<<endl;

    cin>>temp.score;

}

void display(struct stud &temp){

    cout<<"Roll is : "<<temp.roll<<" and score is : "<<temp.score<<endl;

}

int main(){

    struct stud a;

    for(;;){

        cout<<"Enter : \n1 To take input\n2. To display\n3. To modify\n4. To display(global)\n5.To exit\n";

        int c;

        cin>>c;

        switch(c){

            case 1:{

                a.inpt(a);

                break;

            }

            case 2:{

                a.dis(a);

                break;

            }

            case 3:{

                modify(a);

                break;

            }

            case 4:{

                display(a);

                break;

            }

            case 5:{

                cout<<"Exiting ...\n";

                exit(0);

            }

            default:{

                cout<<"Wrong input";

                break;

            }

        }

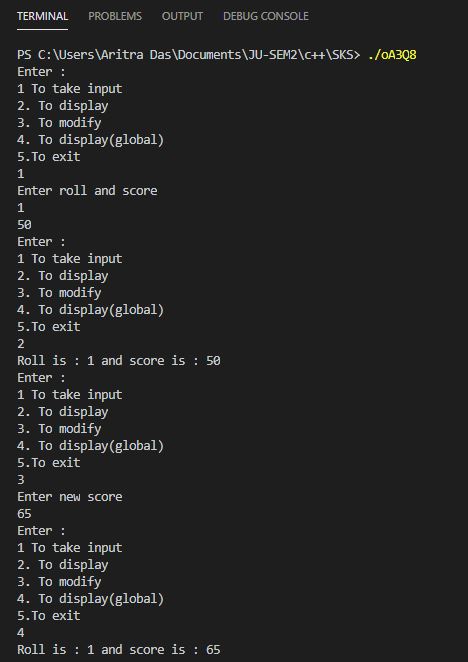
    }

    display(a);

    return 0;

}

Output:



9.

#include <stdio.h>

#include <iostream>

#include <string>

using namespace std;

class TIME{

    private:

    long int inp;

    int hour,min,sec;

    string ap;

    bool pk,ok;

    public:

    void extract(){

        sec=inp%100;

        min=(inp/100)%100;

        hour=inp/10000;

        check();

    }

    void check(){

        if(pk){

            if(hour<0||hour>24||min<0||min>59||sec<0||sec>59){

            cout<<"Wrong Input Program Closing ...";

            exit(0);

            }

        }

        else{

            if(hour<0||hour>12||min<0||min>59||sec<0||sec>59){

            cout<<"Wrong Input Program Closing ...";

            exit(0);

            }

        }

    }

    void i24(){

        cout<<"Enter time in HHMMSS format"<<endl;

        cin>>inp;

        extract();

    }

    void i12(){

        cout<<"Enter only time in HHMMSS format"<<endl;

        cin>>inp;

        cout<<"Enter AM or PM"<<endl;

        cin>>ap;

        extract();

        //check();

        if(ap[0]=='P'||ap[0]=='p')

        hour+=12;

    }

    void o24(){

        cout<<"Time is  "<<hour<<":"<<min<<":"<<sec<<endl;

    }

    void o12(){

        if(hour>12)

        cout<<"Time is  "<<hour-12<<":"<<min<<":"<<sec<<" PM"<<endl;

        else

        cout<<"Time is  "<<hour<<":"<<min<<":"<<sec<<" AM"<<endl;

    }

    void input(){

        cout<<"Enter 1 for 24 hour format entry and 0 for 12 hour format entry: "<<endl;

        cin>>pk;

        if(pk)

        i24();

        else

        i12();

    }

    void output(){

        cout<<"Enter 1 for 24 hour format entry and 0 for 12 hour format output: "<<endl;

        cin>>ok;

        if(ok)

        o24();

        else

        o12();

    }

};

int main(){

    TIME t1;

    for(;;){

        cout<<"Enter : \n1 To take Time input\n2.To show Time output\n3.To exit\n";

        int c;

        cin>>c;

        switch(c){

            case 1:{

                t1.input();

                break;

            }

            case 2:{

                t1.output();

                break;

            }

            case 3:{

                cout<<"Exiting ...\n";

                exit(0);

            }

            default:{

                cout<<"Wrong input";

                break;

            }

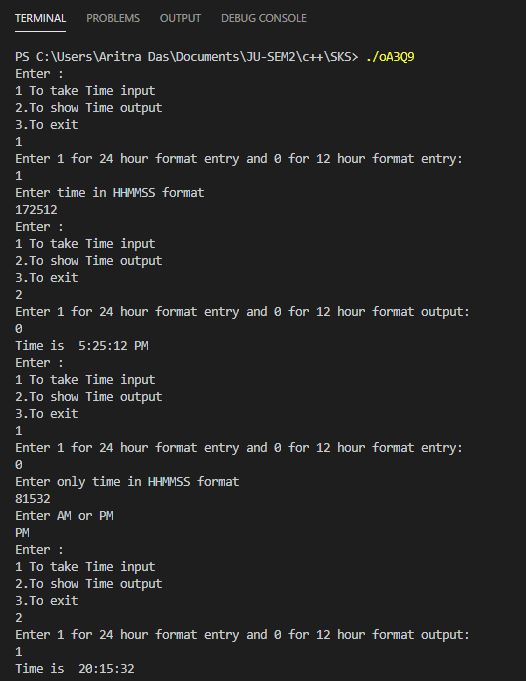
        }

    }

    return 0;

}

Output:



10.

#include <stdio.h>

#include <iostream>

using namespace std;

class STACK{

    private:

    int top=0;

    int arr[100];

    public:

    void push(){

        int data;

        cout<<"Enter data to push"<<endl;

        cin>>data;

        arr[++top]=data;

    }

    void pop(){

        if(top<=-1)

        cout<<"Stack Underflow"<<endl;

        else {

        cout<<"The popped element is "<< arr[top] <<endl;

        top--;

   }

    }

    void show(){

        cout<<"The stack is :"<<endl;

        for(int i=top;i>0;i--)

        cout<<arr[i]<<endl;

    }

};

int main(){

    STACK s1;

    for(;;){

        cout<<"\nEnter \n 1. Push \n 2. Pop \n 3. Show \n 4. Exit"<<endl;

    int a;

    cin>>a;

    switch (a)

    {

        case 1:{

            s1.push();

            break;

        }

        case 2:{

            s1.pop();

            break;

        }

        case 3:{

            s1.show();

            break;

        }

        case 4:{

            exit(0);

            break;

        }

        default:{

            cout<<"Wrong Entry"<<endl;

            break;

        }

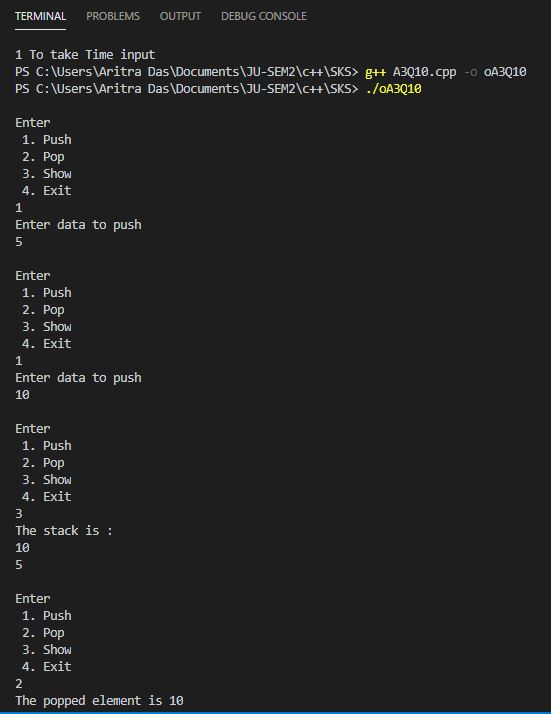
    }

    }

    return 0;

}

Output:



11.

#include <stdio.h>

#include <iostream>

#include <string>

using namespace std;

class APPLICANT{

    int id,score;

    string name;

    public:

    APPLICANT(){

        name = "";

        score = 0;

        id = 0;

    }

    APPLICANT(string nm,int sc,int ids){

        name = nm;

        score = sc;

        id = ids;

    }

    int getId(){

        return id;

    }

    void show(){

        cout<<"\nId of Applicant: "<<id<<endl;

        cout<<"Name of Applicant: "<<name<<endl;

        cout<<"Score of Applicant: "<<score<<endl;

    }

};

class APPSTK{

    private:

    int top=0;

    APPLICANT arr[100];

    public:

    void push(){

        int ida,scr;

        string nam;

        cout<<"Enter name, score, id( enter 0 for default id ) of applicant "<<endl;

        cin>>nam>>scr>>ida;

        ++top;

        if(ida==0){

            if(top==1)

            arr[top]= APPLICANT(nam,scr,1);

            else

            arr[top]= APPLICANT(nam,scr,arr[top-1].getId()+1);

        }

        else

        arr[top]= APPLICANT(nam,scr,ida);

    }

    void show(){

        cout<<"The applicant list is :"<<endl;

        for(int i=top;i>0;i--)

        arr[i].show();

    }

    void find(){

        cout<<"The number of applicants are "<<top<<endl;

    }

};

int main(){

    APPSTK s1;

    for(;;){

        cout<<"\nEnter \n 1. Enter new applicant \n 2. Find number of applicants \n 3. Show all applicants \n 4. Exit"<<endl;

    int a;

    cin>>a;

    switch (a)

    {

        case 1:{

            s1.push();

            break;

        }

        case 2:{

            s1.find();

            break;

        }

        case 3:{

            s1.show();

            break;

        }

        case 4:{

            exit(0);

            break;

        }

        default:{

            cout<<"Wrong Entry"<<endl;

            break;

        }

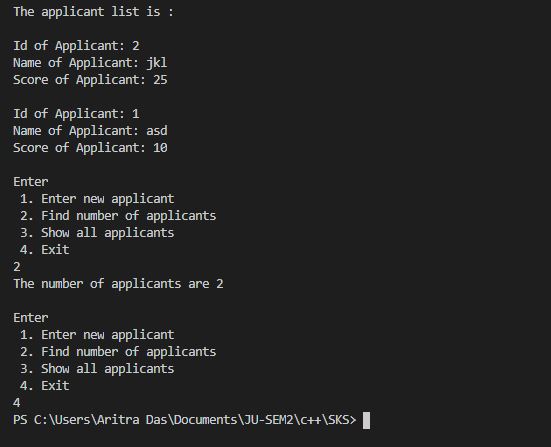
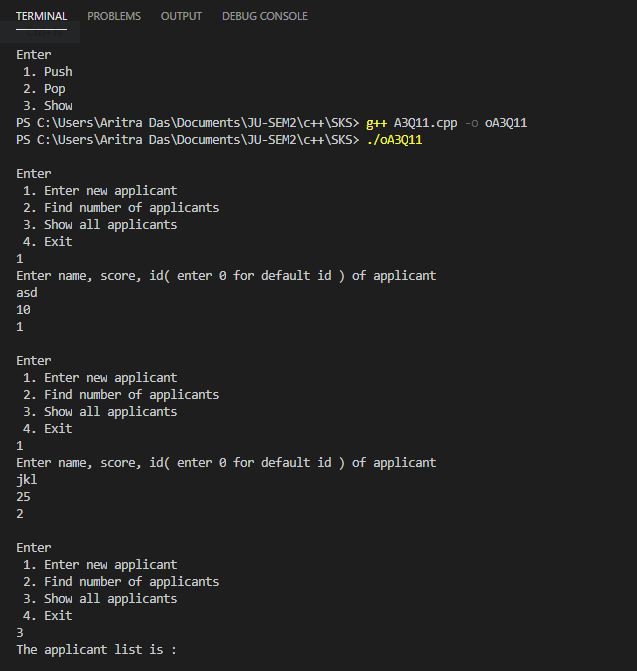
    }

    }

    return 0;

}

Output:



12.

#include <iostream>

#include <string>

using namespace std;

class STUDENT{

    int roll, marks[5];

    string name,course,date;

    public:

    STUDENT(){

        name = "";

        date = "";

        course="";

        roll = 0;

    }

    void admission(){

        cout<<"Enter Name, Roll Number, Course and Date (in DD/MM/YY format)  "<<endl;

        cin>>name>>roll>>course>>date;

    }

    void marksinit(){

        cout<<"Enter Marks for subjects"<<endl;

        for(int i=0;i<5;i++){

            cout<<"Subject "<<i+1<<" :  ";

            cin>>marks[i];

        }

    }

    void marksheet(){

        cout<<"\nRoll Number of Student: "<<roll<<endl;

        cout<<"Name of Student: "<<name<<endl;

        cout<<"Date of Admission: "<<date<<endl;

        cout<<"Enrolled Course: "<<course<<endl;

        cout<<"Marks obtained: "<<endl;

        for(int i=0;i<5;i++)

        cout<<"Subject "<<i+1<<" : "<<marks[i]<<endl;

    }

};

int main(){

    int top=-1;

    STUDENT s1[100];

    for(;;){

        cout<<"\nEnter \n 1. Start a new admission \n 2. Enter marks of the student \n 3. Show the marksheet of current student\n 4. Show the marksheet of all students and total number of students  \n 5. Exit"<<endl;

    int a;

    cin>>a;

    switch (a)

    {

        case 1:{

            s1[++top].admission();

            break;

        }

        case 2:{

            if(top>-1)

            s1[top].marksinit();

            else

            cout<<"No students are present";

            break;

        }

        case 3:{

            if(top>-1)

            s1[top].marksheet();

            else

            cout<<"No students are present";

            break;

        }

        case 4:{

            if(top>-1){

                cout<<"There are "<<top+1<<" students. And their marks are as follows: "<<endl;

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

                s1[i].marksheet();

            }

            //s1[top].marksheet();

            else

            cout<<"No students are present";

            break;

        }

        case 5:{

            exit(0);

            break;

        }

        default:{

            cout<<"Wrong Entry"<<endl;

            break;

        }

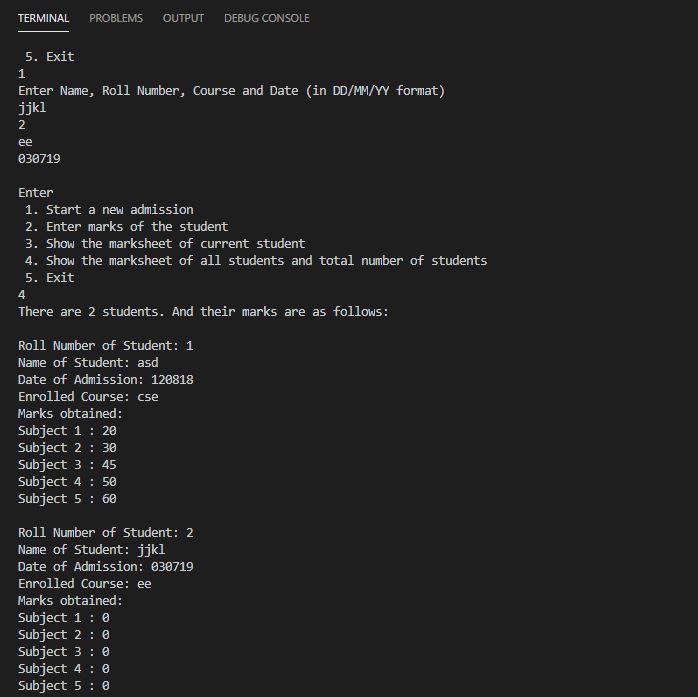
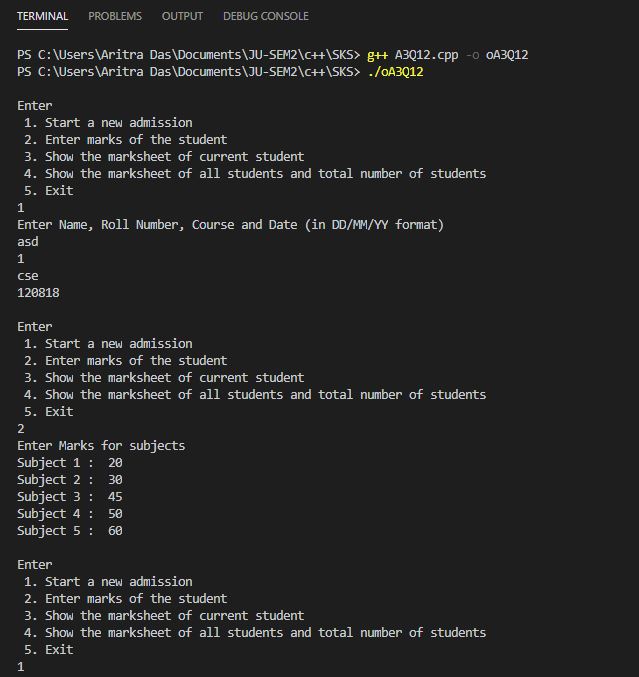
    }

    }

    return 0;

}

Output:



13.

#include <iostream>

#include <string>

using namespace std;

class NODE{

    public:

        int data;

        NODE \*link;

};

class LIST{

    private:

    NODE\* head;

    public:

        LIST(){

            head = NULL;

        }

        void addNodeHead(int x){

            NODE\* temp = new NODE;

            temp->data = x;

            temp->link = head;

            head = temp;

            print();

        }

        void insert(int x, int n){

            NODE\* temp1 = new NODE;

            temp1->data = x;

            temp1->link = NULL;

            if(n==1){

                temp1->link = head;

                head = temp1;

                return;

            }

            NODE\* temp2;

            temp2 = head;

            for(int i =0; i<n-2; i++)

            temp2 = temp2->link;

            temp1->link = temp2->link;

            temp2->link = temp1;

            print();

        }

        void addEnd(int data){

            NODE \*temp = new NODE;

            temp->data = data;

            temp->link = NULL;

            if(head == NULL)

            head = temp;

            else

            {

                NODE \*temp2;

                temp2 = head;

                while(temp2->link != NULL)

                temp2 = temp2->link;

                temp2->link = temp;

            }

            print();

        }

        void deleteN(int n){

            NODE\* temp1;

            temp1 = head;

            for(int i=0;i<n-2;i++)

            temp1 = temp1->link;

            NODE\* temp2 = temp1->link;

            temp1->link = temp2->link;

            delete(temp2);

            print();

        }

        void print(){

            NODE\* temp;

            temp = head;

            while(temp->link!=NULL){

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

                temp=temp->link;

            }

            cout<<temp->data<<" -> ";

            cout<<endl;

        }

};

int main(){

    LIST l;

    for(;;){

        int s;

        cout<<"\nEnter: 1 for Insert At End\n2 for Insert at Head\n3 for deleting at n'th position\n4 for inserting at n'th position\n5 for printing the list\n6 to exit\n ";

        cin>>s;

        switch (s)

        {

            case 1:{

                int data;

                cout<<"Enter data to insert at end\n";

                cin>>data;

                l.addEnd(data);

                break;

            }

            case 2:{

                int data;

                cout<<"Enter data to insert at head\n";

                cin>>data;

                l.addNodeHead(data);

                break;

            }

            case 3:{

                int n;

                cout<<"Enter position to delete\n";

                cin>>n;

                l.deleteN(n);

                break;

            }

            case 4:{

                int data,n;

                cout<<"Enter data to insert\n";

                cin>>data;

                cout<<"Enter position to insert\n";

                cin>>n;

                l.insert(data,n);

                break;

            }

            case 5:{

                cout<<"The list is:\n";

                l.print();

                break;

            }

            case 6:{

                cout<<"Exiting ....";

                exit(1);

                break;

            }

            default:

                cout<<"Invalid Input";

                break;

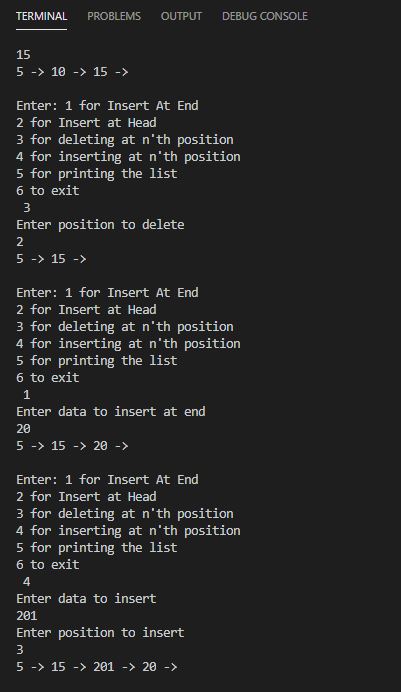
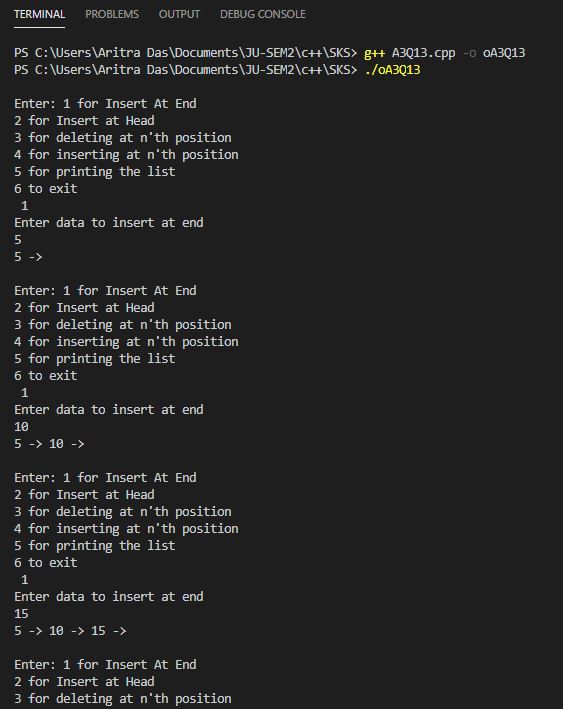
        }

    }

    return 0;

}

Output:



14.

#include<iostream>

#include<string>

using namespace std;

class Item{

    //Item Node initialize

    public:

    int code,rate,quantity;

    string name;

    Item \*next;

};

class ItemList{

    //Item list initialize

    private:

    Item\* head;

    int n;

    public:

    ItemList(){

        head = NULL;

        n=0;

    }

    void insertItem(int code1,string name1,int rate1, int quantity1=0){

        Item\* temp1 = new Item;

        temp1->code = code1;

        temp1->rate = rate1;

        temp1->name = name1;

        temp1->quantity = quantity1;

        temp1->next = NULL;

        if(head == NULL)

        head = temp1;

        else{

            Item\* temp2;

            temp2 = head;

            while(temp2->next!=NULL)

            temp2=temp2->next;

            temp2->next=temp1;

        }

        n++;

        cout<<"Item inserted successfully\n";

    }

    bool checkUnique(int code1){

        Item\* temp ;

        temp = head;

        if(head==NULL)

        return true;

        while(temp!=NULL){

            if(temp->code == code1)

            return false;

            temp = temp->next;

        }

        return true;

    }

    int getRate(int code1){

        Item\* temp ;

        temp = head;

        while(temp!= NULL){

            if(temp->code==code1)

            return temp->rate;

            temp = temp->next;

        }

        return -1;

    }

    int getQuantity(int code1){

        Item\* temp ;

        temp = head;

        while(temp!= NULL){

            if(temp->code==code1)

            return temp->quantity;

            temp = temp->next;

        }

        return -1;

    }

    void updateRate(int code1,int rate1){

        Item\* temp ;

        temp = head;

        while(temp!= NULL){

            if(temp->code==code1){

                temp->rate = rate1;

                cout<<"Rate has been updated\n";

                return;

            }

            temp = temp->next;

        }

        cout<<"Item not found\n";

    }

    void recieve(int code1,int quantity2){

        Item\* temp;

        temp = head;

        while(temp!= NULL){

            if(temp->code==code1){

                temp->quantity += quantity2;

                cout<<"Item recieved. Quantity has been updated to :"<<temp->quantity<< "\n";

                return;

            }

            temp = temp->next;

        }

        cout<<"Item not found";

    }

    void issue(int code1,int quantity2){

        Item\* temp ;

        temp = head;

        while(temp!= NULL){

            if(temp->code==code1){

                if(temp->quantity>=quantity2){

                    temp->quantity -= quantity2;

                    cout<<"Item issued. Quantity has been updated to :"<<temp->quantity<< "\n";

                    return;

                }

                else{

                    cout<<"Items insufficient for issuing \n";

                    return;

                }

            }

            temp = temp->next;

        }

        cout<<"Item not found";

    }

};

int main(){

    ItemList l;

    for(;;){

        cout<<"Enter :\n1 for Adding Item \n2 for getting Rate of an item\n3 for getting Quantity of an item\n4 for updating Rate of an item\n5 for issuing an item\n6 for recieving an item\n7 to Exit\n ";

        int s;

        cin>>s;

        switch(s){

            case 1:{

                cout<<"Enter details of item in order(code,name,rate,quantity)\n";

                Item \*temp= new Item;

                cin>>temp->code>>temp->name>>temp->rate>>temp->quantity;

                if(!l.checkUnique(temp->code)){

                    cout<<"Item code exists";

                    break;

                }

                else{

                    l.insertItem(temp->code,temp->name,temp->rate,temp->quantity);

                }

                delete(temp);

                break;

            }

            case 2:{

                cout<<"Enter Item Code to get rate\n";

                int x;

                cin>>x;

                if(l.getRate(x)==-1){

                    cout<<"Item does not exist in the list\n";

                    break;

                }

                else{

                    cout<<"The rate of the item is : "<<l.getRate(x)<<endl;

                    break;

                }

            }

            case 3:{

                cout<<"Enter Item Code to get quantity\n";

                int x;

                cin>>x;

                if(l.getQuantity(x)==-1){

                    cout<<"Item does not exist in the list\n";

                    break;

                }

                else{

                    cout<<"The quantity of the item is : "<<l.getQuantity(x)<<endl;

                    break;

                }

            }

            case 4:{

                cout<<"Enter Item code and rate to update\n";

                int a,b;

                cin>>a>>b;

                l.updateRate(a,b);

                break;

            }

            case 5:{

                cout<<"Enter Item code and quantity to issue\n";

                int a,b;

                cin>>a>>b;

                l.issue(a,b);

                break;

            }

            case 6:{

                cout<<"Enter Item code and quantity to recieve\n";

                int a,b;

                cin>>a>>b;

                l.recieve(a,b);

                break;

            }

            case 7:{

                cout<<"Exiting program\n";

                exit(1);

            }

            default:{

                cout<<"Wrong Input";

                break;

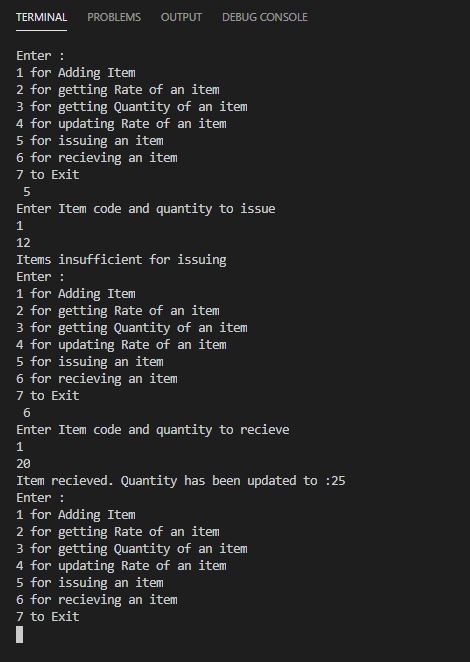
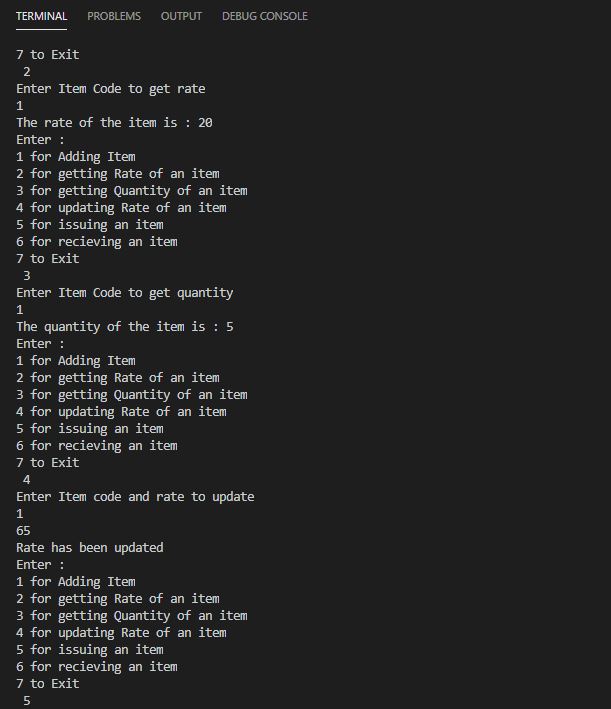
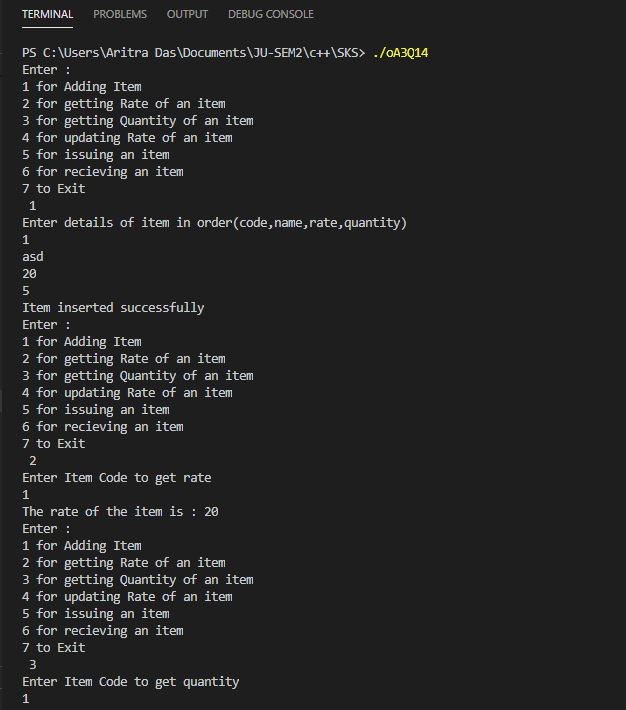
            }

        }

    }

}

Output:



15.

#include<iostream>

#include<string>

using namespace std;

class BALANCE{

    public:

    int accNum;

    unsigned long int balance;

    string date;

    BALANCE\* next;

    void setBal(unsigned long int x, char type){

        if(type=='W')

        balance -= x;

        else if(type=='D')

        balance+=x;

    }

};

class TRANSACTION{

    private:

    BALANCE \* head;

    int n=0;

    public:

    TRANSACTION(){

        head = NULL;

    }

    void display(BALANCE\* temp){

        cout<<"\nAccount Number: "<<temp->accNum<<"\nBalance: "<<temp->balance<<"\nDate of last update: "<<temp->date;

    }

    void initAcc(){

        cout<<"Enter Account Number to open an account\n";

        int acc;

        cin>> acc;

        BALANCE \*temp = new BALANCE;

        temp->accNum = acc;

        temp->balance = 0;

        display(temp);

        if(head == NULL)

        head = temp;

        else{

            BALANCE \* temp2;

            temp2 = head;

            while(temp2!=NULL)

            temp2= temp2->next;

            temp2->next = temp;

        }

        n++;

    }

    void withdraw(){

        cout<<"Enter account number, amount and date \n";

        int acc;

        unsigned long int amount;

        string date;

        char type;

        cin>>acc>>amount>>date;

        type='W';

        BALANCE \* temp;

        temp = head;

        while(temp!=NULL){

            if(temp->accNum==acc){

                if(temp->balance<amount){

                    cout<<"Insufficient Balance\n";

                    return;

                }

                else{

                    temp->setBal(amount,type);

                    temp->date = date;

                    display(temp);

                    return;

                }

            }

            temp=temp->next;

        }

        cout<<"Account not found\n";

        return;

    }

    void deposit(){

        cout<<"Enter account number, amount and date \n";

        int acc;

        unsigned long int amount;

        string date;

        char type;

        cin>>acc>>amount>>date;

        type='D';

        BALANCE \* temp;

        temp = head;

        while(temp!=NULL){

            if(temp->accNum==acc){

                temp->setBal(amount,type);

                temp->date = date;

                display(temp);

                return;

            }

        }

        cout<<"Account not found\n";

        return;

    }

};

int main(){

    TRANSACTION t;

    for(;;){

        int s;

        cout<<"\nEnter : \n1 to open an Account\n2 to Withdraw money\n3 to Deposit money\n4 to Exit\n";

        cin>>s;

        switch(s){

            case 1:{

                t.initAcc();

                break;

            }

            case 2:{

                t.withdraw();

                break;

            }

            case 3:{

                t.deposit();

                break;

            }

            case 4:{

                cout<<"Exiting...";

                exit(1);

            }

            default:{

                cout<<"Wrong Input\n";

                break;

            }

        }

    }

    return 0;

}