Name : Aritra Das

Roll No: 75

Assignment 4

1.

#include<iostream>

#include<string>

using namespace std;

class Account{

    public:

    int acNo;

    unsigned long int bnc;

    Account\* next;

    friend void withdraw(Account\*);

};

void withdraw(Account \*a){

        cout<<"Enter money to withdraw: \n";

        long int bnc1;

        cin>>bnc1;

        if(a->bnc>=bnc1){

            a->bnc-=bnc1;

            cout<<"Remaining balance: "<<a->bnc<<endl;

        }

        else

        cout<<"Unsufficient balance\n";

}

class AcList{

    private:

    Account\* head;

    public:

    AcList(){

        head = NULL;

    }

    void push(int a,int b){

        Account\* temp = new Account;

        temp->acNo = a;

        temp->bnc = b;

        temp->next = NULL;

        if(head == NULL)

        head = temp;

        else{

            Account\* temp1;

            temp1 = head;

            while(temp1->next=NULL){

                if(!checkUnique(a)){

                    cout<<"Account already exists\n";

                    return;

                }

                temp1=temp1->next;

            }

            temp1->next = temp;

        }

    }

    bool checkUnique(int acNo1){

        Account\* temp;

        temp = head;

        if(head==NULL)

        return true;

        while(temp!=NULL){

            if(temp->acNo == acNo1)

            return false;

            temp = temp->next;

        }

        return true;

    }

    void display(){

        Account\* temp ;

        temp = head;

        while(temp!=NULL){

            cout<<"Account Number: "<<temp->acNo<<" Balance: "<<temp->bnc<<endl;

            temp = temp->next;

        }

    }

    void with(int acNo1){

        Account\* temp ;

        temp = head;

        while(temp!=NULL){

            if(temp->acNo==acNo1)

            withdraw(temp);

            temp = temp->next;

        }

        cout<<"Acount does not exist";

    }

};

int main(){

    AcList a1;

    for(;;){

        cout<<"Enter : \n1. To open an account\n2. To withdraw money\n3. Display all acoounts \n4. Exit\n";

        int s;

        cin>>s;

        switch(s){

            case 1:{

                cout<<"Enter account number and balance\n";

                int a;

                unsigned long int b;

                cin>>a>>b;

                a1.push(a,b);

                break;

            }

            case 2:{

                cout<<"Enter account number for withdrawl:\n";

                int a;

                cin>>a;

                a1.with(a);

                break;

            }

            case 3:{

                cout<<"List :\n";

                a1.display();

                break;

            }

            case 4:{

                cout<<"Exiting...\n";

                exit(0);

            }

            default:{

                cout<<"wrong input \n";

                break;

            }

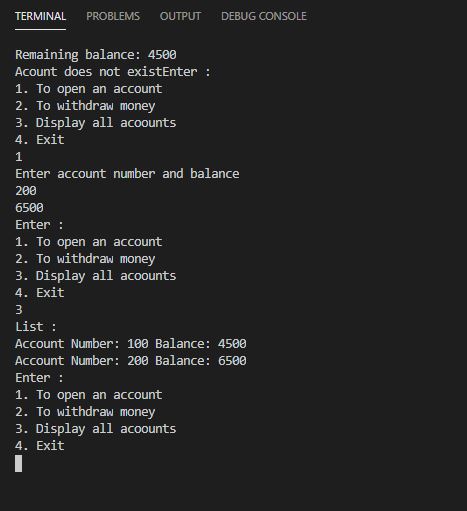
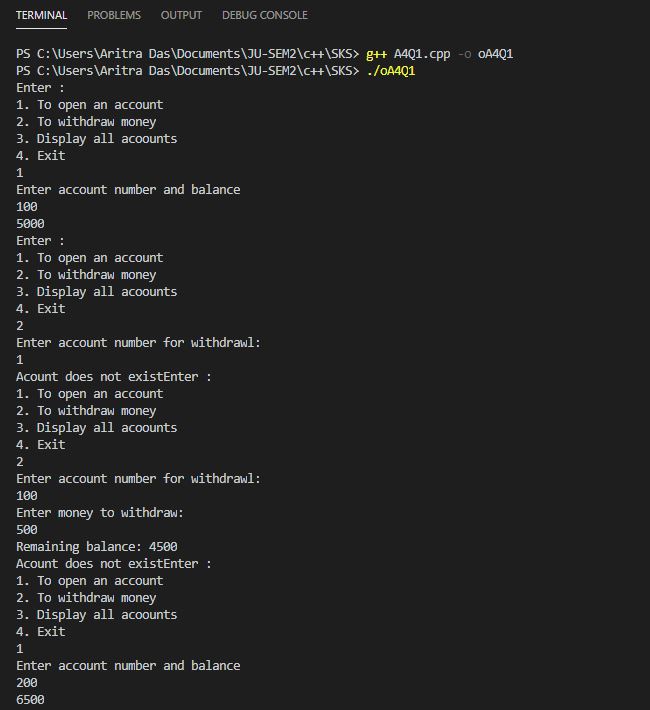
        }

    }

    return 0;

}

Output:



2.

#include<iostream>

using namespace std;

class COMPLEX{

    int real;

    int img;

    public:

        COMPLEX(int r = 0, int i =0)  {real = r;   img = i;}

        void setV()

        {

            cin>>real>>img;

        }

        COMPLEX operator+(COMPLEX const &c)

        {

            COMPLEX co;

            co.real=real+c.real;

            co.img=img+c.img;

            return co;

        }

        COMPLEX operator-(COMPLEX const &c)

        {

            COMPLEX com;

            com.real=real-c.real;

            com.img=img-c.img;

            return com;

        }

        void print()

        {

            cout<<real<<"         i"<<img<<"\n";

        }

};

int main()

{

    COMPLEX f1,f2,f3;

while(1)

{

    cout<<"Enter :\n1 for addition\n2 for substraction\n3To quit\n";

    int ch;

    cin>>ch;

    switch(ch)

    {

        case 3: {cout<<"Exiting ...\n";

        exit(0);

        break;}

        case 1: {cout<<"Enter real and img parts\n";

         f1.setV();

        cout<<"Enter second real and img parts\n";

        f2.setV();

        f3=f1+f2;

        f3.print();

        break;}

        case 2:{

        cout<<"Enter real and img parts\n";

         f1.setV();

        cout<<"Enter second real and img partsr\n";

         f2.setV();

         f3=f1-f2;

        f3.print();

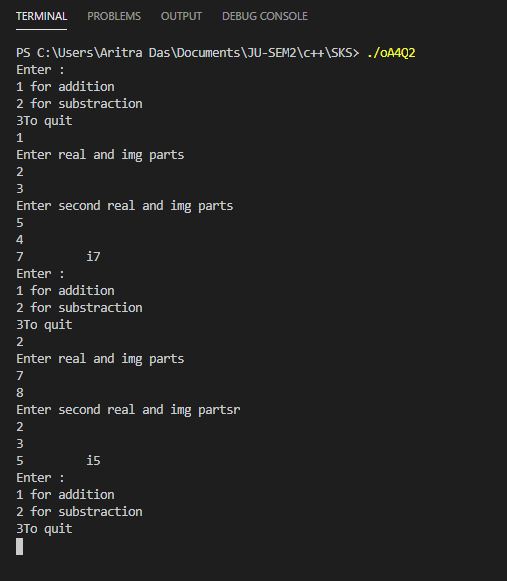
        break;}

    }

}

}

Output:



3.

#include<iostream>

#include<string>

using namespace std;

class ARRAY{

    private:

    int n;

    int \*arr;

    public:

    // ARRAY(){

    //     arr = new int (0);

    // }

    ARRAY(int s=0,int x=0){

        if(s>0)

        {

            n=s;

            arr=new int[n];

            for(int i=0;i<n;i++)

                arr[i]=x;

        }

        else{

            n=0;

            arr=NULL;

        }

    }

    ARRAY(const ARRAY &a){

        n=a.n;

        arr = new int(n);

        for(int i=0;i<a.n;i++)

        arr[i] = a.arr[i];

    }

    ARRAY(int \*arr1){

        arr = arr1;

    }

    void operator= (const int\* p)

    {

        for(int i=0;i<n;i++)

            arr[i]=p[i];

    }

    void operator= (const ARRAY &a)

    {

        n=a.n;

        arr=new int[a.n];

        for(int i=0;i<n;i++)

            arr[i]=a.arr[i];

    }

    ARRAY operator+ (const ARRAY &c){

        ARRAY a1(c.n,0);

        for(int i =0 ; i<c.n;i++)

        a1.arr[i] = arr[i] + c.arr[i];

        return a1;

    }

    int& operator[](int i){

        return arr[i];

    }

    ARRAY operator \*(int x)

    {

        ARRAY c(n);

        for(int i=0;i<n;i++)

            c.arr[i]=x\*arr[i];

        return c;

    }

};

int main(){

    ARRAY b(6,5);

    for(int i=0;i<6;i++)

    cout<<b[i]<<" ";

    cout<<endl;

    for(;;){

        cout<<"Enter : \n1. To create an Array object\n2. To create an Array object with another Array object\n3. To create an Array object with a primitive array\n4. To add two Array objects\n5. To multiply Array object with an integer\n6. To equate two Array objects\n7. To display all elements of an Array object using the subscript operator\n8. To exit";

        int c;

        cin>>c;

        switch(c){

            case 1:{

                int a,b;

                cout<<"Enter size and value for assignment\n";

                cin>>a>>b;

                ARRAY a1(a,b);

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

                for(int i=0;i<a;i++)

                cout<<a1[i]<<" ";

                cout<<endl;

                break;

            }

            case 2:{

                int a,b;

                cout<<"Enter size and value for assignment of first Array object\n";

                cin>>a>>b;

                ARRAY b1(a,b);

                cout<<"The first Array object is: \n";

                for(int i=0;i<a;i++)

                cout<<b1[i]<<" ";

                cout<<endl;

                cout<<"The second copied Array object is: \n";

                ARRAY c(a);

                c = b;

                for(int i=0;i<a;i++)

                cout<<c[i]<<" ";

                cout<<endl;

                break;

            }

            case 3:{

                int a,b;

                cout<<"Enter size and value for assignment of primitive array\n";

                cin>>a>>b;

                int \*x;

                x = new int(a);

                for(int i=0;i<a;i++)

                x[i] = b;

                cout<<"The primitive array is: \n";

                for(int i=0;i<a;i++)

                cout<<x[i];

                cout<<endl;

                ARRAY e(x);

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

                for(int i=0;i<a;i++)

                cout<<e[i]<<" ";

                cout<<endl;

                break;

            }

            case 4:{

                int a,b;

                cout<<"Enter size and value for assignment of first Array object\n";

                cin>>a>>b;

                ARRAY b1(a,b);

                cout<<"The first Array object is: \n";

                for(int i=0;i<a;i++)

                cout<<b1[i]<<" ";

                int x,y;

                cout<<"Enter size and value for assignment of second Array object\n";

                cin>>x>>y;

                ARRAY b2(x,y);

                cout<<"The second Array object is: \n";

                for(int i=0;i<a;i++)

                cout<<b2[i]<<" ";

                if(x!=a){

                    cout<<"Array object lengths are not same, please enter same length\n";

                    break;

                }

                ARRAY d = b1+b2;

                cout<<"The added array objects result is:\n";

                for(int i=0;i<a;i++)

                cout<<d[i]<<" ";

                cout<<endl;

                break;

            }

            case 5:{

                int a,b,c;

                cout<<"Enter size and value for assignment of first Array object\n";

                cin>>a>>b;

                ARRAY b1(a,b);

                cout<<"The first Array object is: \n";

                for(int i=0;i<a;i++)

                cout<<b1[i]<<" ";

                cout<<"Enter integer to multiply object with:\n";

                cin>>c;

                ARRAY f(6);

                f = b1\*c;

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

                for(int i=0;i<a;i++)

                cout<<f[i]<<" ";

                cout<<endl;

                break;

            }

            case 6:{

                int a,b;

                cout<<"Enter size and value for assignment of first Array object\n";

                cin>>a>>b;

                ARRAY b1(a,b);

                cout<<"The first Array object is: \n";

                for(int i=0;i<a;i++)

                cout<<b1[i]<<" ";

                cout<<"The equated Array object is\n";

                ARRAY c(a);

                c = b1;

                for(int i=0;i<a;i++)

                cout<<c[i]<<" ";

                cout<<endl;

                break;

            }

            case 7:{

                int a,b;

                cout<<"Enter size and value for assignment of first Array object\n";

                cin>>a>>b;

                ARRAY b1(a,b);

                cout<<"The Array object is: \n";

                for(int i=0;i<a;i++)

                cout<<b1[i]<<" ";

                break;

            }

            case 8:{

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

                exit(0);

                break;

            }

            default:{

                cout<<"Wrong input";

                break;

            }

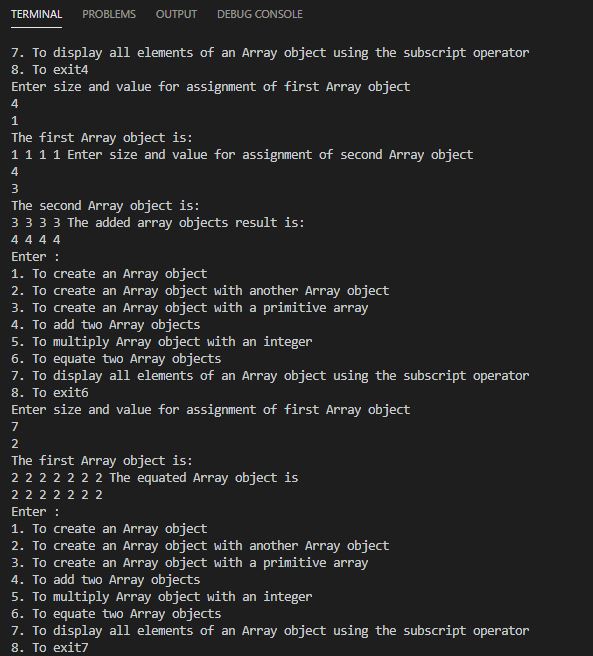
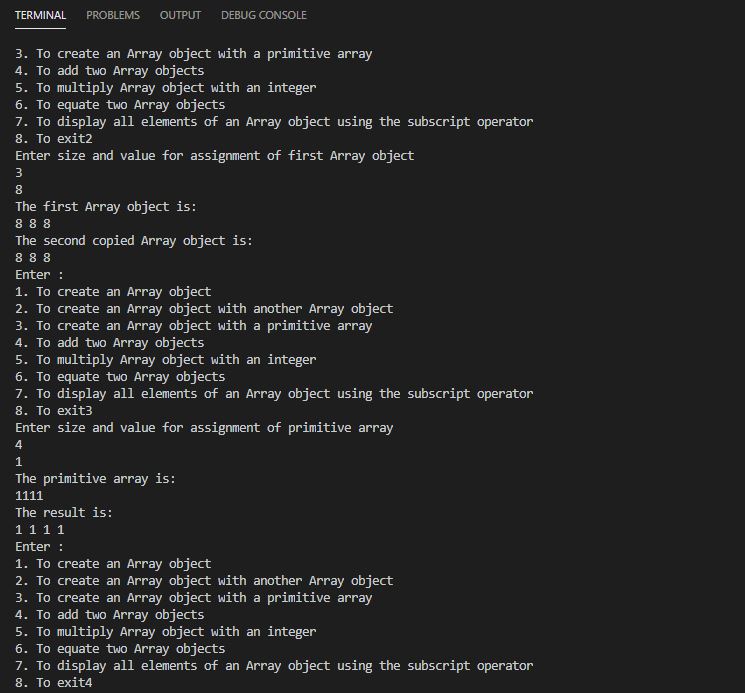
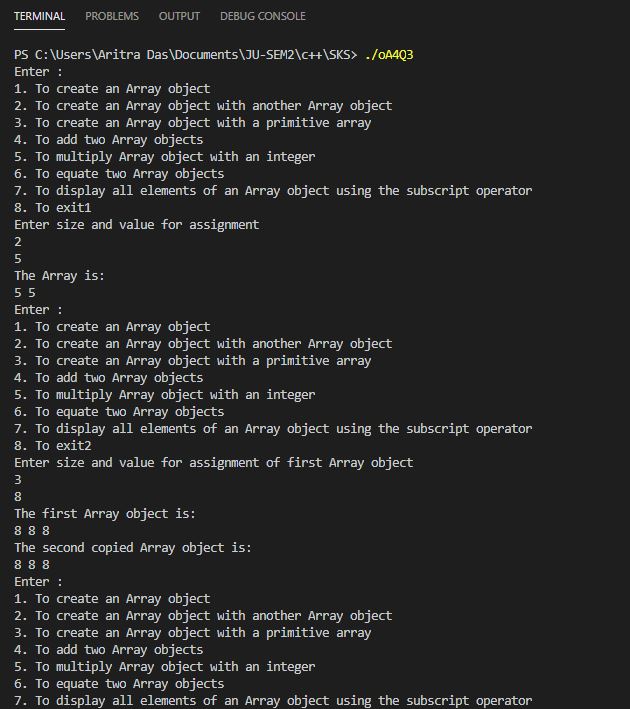
        }

    }

    return 0;

}

Output:



4.

#include <iostream>

using namespace std;

class STRING{

    int sz;

    char \*data;

    public:

        //constructor

        STRING(int l=0, char c='\0'){

            if(l>0){

                sz=l;

                data=new char[sz+1];

                data[sz]='\0';

                for(int i=0;i<l;i++)

                    data[i]=c;

            }

            else{

                sz=0;

                data=NULL;

            }

        }

        //copy constructor

        STRING(const STRING& s){

            sz=s.sz;

            if(sz==0)

                data=NULL;

            else{

                //delete data;

                data=new char[sz+1];

                data[sz]='\0';

                for(int i=0;i<sz;i++)

                    data[i]=s.data[i];

            }

        }

        //assignment operator

        void operator =(const STRING& s){

            if(sz!=0)

            delete data;

            sz=s.sz;

            if(sz==0)

                data=NULL;

            else{

                data=new char[sz+1];

                data[sz]='\0';

                for(int i=0;i<sz;i++)

                    data[i]=s.data[i];

            }

        }

        //concatenation

        STRING operator +(const STRING& s){

            STRING res;

            res.sz=sz+s.sz;

            res.data=new char[res.sz+1];

            for(int i=0;i<sz;i++)

                res.data[i]=data[i];

            for(int i=0;i<s.sz;i++)

                res.data[sz+i]=s.data[i];

            res.data[res.sz]='\0';

            return res;

        }

        //comparing

        bool operator ==(const STRING& s){

            if(sz!=s.sz)

                return false;

            if(sz==0 && s.sz==0)

                return true;

            for(int i=0;i<sz;i++){

                if(data[i]!=s.data[i])

                    return false;

            }

            return true;

        }

        //< relational operator

        bool operator <(const STRING& s){

            if(sz==0 && s.sz==0)

                return false;

            int min=(sz<s.sz)?sz:s.sz;

            for(int i=0;i<min;i++){

                if(data[i]<s.data[i])

                    return true;

                else if(data[i]>s.data[i])

                    return false;

            }

            if(sz==s.sz)

                return false; //both are identical

            if(sz<s.sz)

                return true; //first string has fewer characters

            if(sz>s.sz)

                return false; // first string has more characters

        }

        //> relational operator

        bool operator >(const STRING& s){

            if(sz==0 && s.sz==0)

                return false;

            int min=(sz<s.sz)?sz:s.sz;

            for(int i=0;i>min;i++){

                if(data[i]>s.data[i])

                    return true;

                else if(data[i]<s.data[i])

                    return false;

            }

            //if all prev characters are equal

            if(sz==s.sz)

                return false; //identical

            if(sz>s.sz)

                return true; // first string has more characters

            if(sz<s.sz)

                return false; // first string has less characters

        }

        friend istream& operator >>(istream &in, STRING &s);

        friend ostream& operator <<(ostream &o, STRING const &s);

        //destructor

        ~STRING(){

            if(data!=NULL)

                delete data;

        }

};

//for taking string as input

istream& operator >>(istream& in, STRING& s){

    char ch;

    cout<<"Enter length of string : ";

    in>>s.sz;

    char\* data=new char[s.sz+1];

    data[s.sz]='\0';

    cout<<"Enter string : ";

    for(int i=0;i<s.sz;i++){

        cin>>ch;

        data[i]=ch;

    }

    s.data=data;

    return in;

}

//for displaying string

ostream& operator <<(ostream &o, STRING const &s){

    for(int i=0;i<s.sz;i++)

        cout<<s.data[i];

    return o;

}

int main(){

    STRING s1, s2, s3;

    int ch,c;

    char s;

    do{

        cout<<"\nEnter choice\n1. Add two strings\n2. Compare two strings\n3. Copy of a string\n4. Exit\n\n";

        cin>>ch;

        switch(ch){

            case 1:

                cout<<"Enter string 1\n";

                cin>>s1;

                cout<<"Enter string 2\n";

                cin>>s2;

                s3=s1+s2;

                                cout<<"\n Result :\n";

                cout<<s1<<" + "<<s2<<" = "<<s3<<"\n";

                break;

            case 2:

                cout<<"Enter first string\n";

                cin>>s1;

                cout<<"Enter second string\n";

                cin>>s2;

                do{

                    cout<<"\t1. <\n";

                    cout<<"\t2. ==\n";

                    cout<<"\t3. >\n";

                    cout<<"\t\tEnter choice:";

                    cin>>c;

                    switch(c){

                        case 1: //s1 less than s2 checking

                            cout<<s1<<" < "<<s2<<" = "<<(s1<s2)<<"\n";

                            break;

                        case 2: //Equality

                            cout<<s1<<" == "<<s2<<" = "<<(s1==s2)<<"\n";

                            break;

                        case 3://s1 greater than s2 checking

                            cout<<s1<<" > "<<s2<<" = "<<(s1>s2)<<"\n";

                            break;

                        default://wrong checking

                            cout<<"Wrong choice. Enter again!\n";

                            s='y';

                            continue;

                    }

                    cout<<"Do you want to continue with the same strings? Enter F to change strings, N to stop comparing, and any other key to continue :\n";

                    cin>>s;

                    if(s=='f' || s=='F'){

                        cout<<"Which string do you want to change? 1 or 2\n";

                        do{

                            cin>>c;

                            if(c==1){

                                cout<<"Enter string 1\n";

                                cin>>s1;

                            }

                            else if(c==2){

                                cout<<"Enter string 2\n";

                                cin>>s2;

                            }

                            else{

                                cout<<"Wrong choice...Enter again\n";

                            }

                        }while(c!=1 && c!=2);

                    }

                }while(s!='n' && s!='N');

                break;

            case 3:

                cout<<"Displaying value of string 1 at this moment : "<<s1

                    <<"\nEnter string to copy : \n";

                cin>>s2;

                cout<<"Displaying value of string s2 : "<<s2<<endl;

                s1=s2;

                cout<<"Displaying new value of string 1 : "<<s1<<endl;

                break;

            case 4:

                cout<<"Exiting ... ";

                break;

            default:

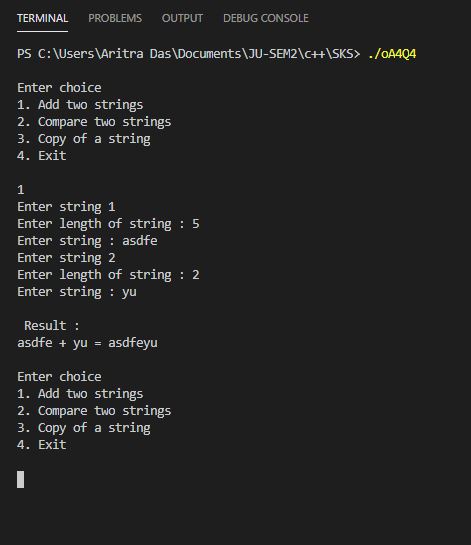
                cout<<"Wrong input\n";

        }

    }while(ch!=4);

}

Output:



5.

#include <iostream>

using namespace std;

class STRINGT{

    int sz;

    int\* refer;

    char \*data;

    public:

        //constructor

        STRINGT(int l=0, char c='\0'){

            if(l>0){

                sz=l;

                data=new char[sz+1];

                data[sz]='\0';

                for(int i=0;i<l;i++)

                    data[i]=c;

            }

            else{

                sz=0;

                data=NULL;

            }

            refer=new int;

            \*refer=1;

            cout<<"refererence to memory block (cons ):"<<refer<<"="<<\*refer<<endl;

        }

        //copy constructor

        STRINGT(const STRINGT& s){

            data=s.data;

            sz=s.sz;

            refer=s.refer;

            \*refer=\*refer+1;

            cout<<"reference to memory block (copy cons ):"<<refer<<"="<<\*refer<<endl;

        }

        //assignment operator

        void operator =(const STRINGT& s){

            data=s.data;

            sz=s.sz;

            refer=s.refer;

            \*refer=\*refer+1;

            cout<<"reference to memory block (op ):"<<refer<<"="<<\*refer<<endl;

        }

        //concatenation

        STRINGT operator +(const STRINGT& s){

            STRINGT res;

            res.sz=sz+s.sz;

            res.data=new char[res.sz+1];

            for(int i=0;i<sz;i++)

                res.data[i]=data[i];

            for(int i=0;i<s.sz;i++)

                res.data[sz+i]=s.data[i];

            res.data[res.sz]='\0';

            cout<<res.data<<endl;

            return res;

        }

        //equality checker

        bool operator ==(const STRINGT& s){

            if(sz!=s.sz)

                return false;

            if(sz==0 && s.sz==0)

                return true;

            for(int i=0;i<sz;i++){

                if(data[i]!=s.data[i])

                    return false;

            }

            return true;

        }

        //< relational operator

        bool operator <(const STRINGT& s){

            if(sz==0 && s.sz==0)

                return false;

            int min=(sz<s.sz)?sz:s.sz;

            for(int i=0;i<min;i++){

                if(data[i]<s.data[i])

                    return true;

                else if(data[i]>s.data[i])

                    return false;

            }

            //if all prev characters are equal

            if(sz==s.sz)

                return false; //since both are identical

            if(sz<s.sz)

                return true; //since first string has less characters so str1<str2

            if(sz>s.sz)

                return false; ////since first string has more characters so str1>str2

        }

        //> relational operator

        bool operator >(const STRINGT& s){

            if(sz==0 && s.sz==0)

                return false;

            int min=(sz<s.sz)?sz:s.sz;

            for(int i=0;i>min;i++){

                if(data[i]>s.data[i])

                    return true;

                else if(data[i]<s.data[i])

                    return false;

            }

            //if all prev characters are equal

            if(sz==s.sz)

                return false; //since both are identical

            if(sz>s.sz)

                return true; //since first string has more characters so str1>str2

            if(sz<s.sz)

                return false; ////since first string has less characters so str1<str2

        }

        friend istream& operator >>(istream &in, STRINGT &s);

        friend ostream& operator <<(ostream &o, STRINGT const &s);

        //destructor

        ~STRINGT(){

            \*refer=\*refer-1;

            cout<<"reference to memory block (=des ):"<<refer<<"="<<\*refer<<endl;

            if(\*refer==0){

                if(sz!=0)

                    delete data;

            }

        }

};

//for taking string as input

istream& operator >>(istream& in, STRINGT& s){

    cout<<"Enter length of string : ";

    in>>s.sz;

    char\* data=new char[s.sz+1];

    data[s.sz]='\0';

    cout<<"Enter string : ";

    for(int i=0;i<s.sz;i++){

        cin>>data[i];

    }

    s.data=data;

    cout<<s.data<<endl;

    return in;

}

//for displaying string

ostream& operator <<(ostream &o, STRINGT const &s){

    for(int i=0;i<s.sz;i++)

        cout<<s.data[i];

    return o;

}

int main(){

    STRINGT s1, s2, s3;

    int ch,c;

    char s;

    do{

        cout<<"\nEnter choice\n1. Add two strings\n2. Compare two strings\n3. Copy of a string\n4. Exit\n\n";

        cin>>ch;

        switch(ch){

            case 1:

                cout<<"Enter string 1\n";

                cin>>s1;

                cout<<"Enter string 2\n";

                cin>>s2;

                s3=s1+s2;

                cout<<s1<<" + "<<s2<<" = "<<s3<<"\n";

                break;

            case 2:

                cout<<"Enter first string\n";

                cin>>s1;

                cout<<"Enter second string\n";

                cin>>s2;

                do{

                    cout<<"1. <\n";

                    cout<<"2. ==\n";

                    cout<<"3. >\n";

                    cout<<"Enter choice\n";

                    cin>>c;

                    switch(c){

                        case 1:

                            cout<<s1<<" < "<<s2<<" = "<<(s1<s2)<<"\n";

                            break;

                        case 2:

                            cout<<s1<<" == "<<s2<<" = "<<(s1==s2)<<"\n";

                            break;

                        case 3:

                            cout<<s1<<" > "<<s2<<" = "<<(s1>s2)<<"\n";

                            break;

                        default:

                            cout<<"Wrong choice. Enter again!\n";

                            s='y';

                            continue;

                    }

                    cout<<"Do you want to continue with the same strings? Press F to change strings, N to stop comparing, and any other key to continue\n";

                    cin>>s;

                    if(s=='f' || s=='F'){

                        cout<<"Which string do you want to change? 1 or 2\n";

                        do{

                            cin>>c;

                            if(c==1){

                                cout<<"Enter string 1\n";

                                cin>>s1;

                            }

                            else if(c==2){

                                cout<<"Enter string 2\n";

                                cin>>s2;

                            }

                            else{

                                cout<<"Wrong input\n";

                            }

                        }while(c!=1 && c!=2);

                    }

                }while(s!='n' && s!='N');

                break;

            case 3:

                cout<<"Displaying value of string 1  : "<<s1<<"\nEnter string to copy : \n";

                cin>>s2;

                cout<<"Displaying value of string s2 : "<<s2<<endl;

                s1=s2;

                cout<<"Displaying new value of string 1 : "<<s1<<endl;

                break;

            case 4:

                cout<<"Exiting\n";

                break;

            default:

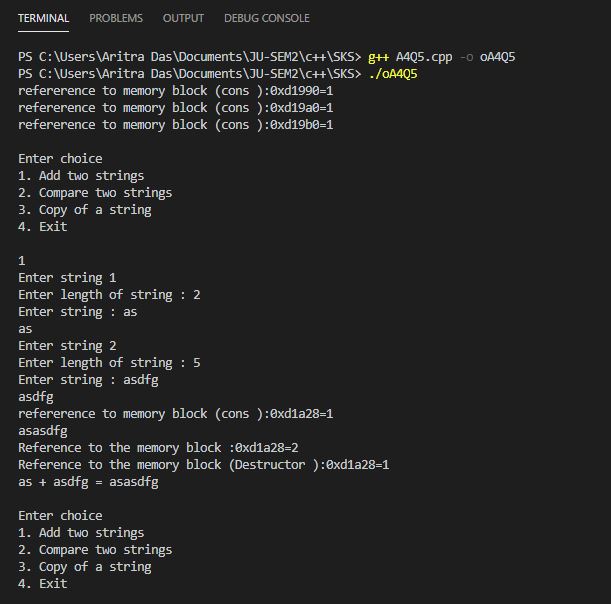
                cout<<"Wrong input\n";

        }

    }while(ch!=4);

}

Output:



6.

#include<iostream>

#include<string>

#include<vector>

using namespace std;

class Student{

    public:

    int roll,phNo;

    string name;

    Student\* next;

    vector<int> sb1;

};

class StuList{

    private:

    Student\* head;

    public:

    StuList(){

        head = NULL;

    }

    void push(int x,string y,int z){

        Student \*temp = new Student;

        temp->roll = x;

        temp->phNo = z;

        temp->name = y;

        temp ->next = NULL;

        if(head == NULL)

        head = temp;

        else{

            Student \*temp1;

            temp1= head;

            while(temp1->next!=NULL)

            temp1=temp1->next;

            temp1->next = temp;

        }

    }

    void display(){

        Student \*temp1;

        temp1= head;

        while(temp1!=NULL){

            cout<<"Roll No : "<<temp1->roll<<" PhNo: "<<temp1->phNo<<" Name: "<<temp1->name<<endl;

            temp1=temp1->next;

        }

    }

    void add(int rll,int code){

        Student \*temp1;

        temp1= head;

        while(temp1!=NULL){

            if(temp1->roll==rll)

            break;

            temp1=temp1->next;

        }

        temp1->sb1.push\_back(code);

    }

    vector<int> disSub(int rll){

        Student \*temp1;

        temp1= head;

        while(temp1!=NULL){

            if(temp1->roll==rll)

            break;

            temp1=temp1->next;

        }

        return temp1->sb1;

    }

    void display(int rll){

        Student \*temp1;

        temp1= head;

        while(temp1!=NULL){

            if(temp1->roll==rll){

                cout<<"Roll No : "<<temp1->roll<<" Name: "<<temp1->name<<" PhNo: "<<temp1->phNo<<endl;

                break;

            }

            temp1=temp1->next;

        }

    }

};

class Subject{

    public:

    int code;

    string name;

    Subject\* next;

    vector<int> sl1;

};

class SubList{

    private:

    Subject\* head;

    public:

    SubList(){

        head = NULL;

    }

    void push(int x,string y){

        Subject \*temp = new Subject;

        temp->code = x;

        temp->name = y;

        temp ->next = NULL;

        if(head == NULL)

        head = temp;

        else{

            Subject \*temp1;

            temp1= head;

            while(temp1->next!=NULL)

            temp1=temp1->next;

            temp1->next = temp;

        }

    }

    void display(){

        Subject \*temp1;

        temp1= head;

        while(temp1!=NULL){

            cout<<"Code No : "<<temp1->code<<" Name: "<<temp1->name<<endl;

            temp1=temp1->next;

        }

    }

    void display(int cde){

        Subject \*temp1;

        temp1= head;

        while(temp1!=NULL){

            if(temp1->code==cde){

                cout<<"Code No : "<<temp1->code<<" Name: "<<temp1->name<<endl;

                break;

            }

            temp1=temp1->next;

        }

    }

    void add(int cde, int roll){

        Subject \*temp1;

        temp1= head;

        while(temp1!=NULL){

            if(temp1->code==cde)

            break;

            temp1=temp1->next;

        }

        temp1->sl1.push\_back(roll);

    }

    vector<int> disStu(int cde){

        Subject \*temp1;

        temp1= head;

        while(temp1!=NULL){

            if(temp1->code==cde)

            break;

            temp1=temp1->next;

        }

        return temp1->sl1;

    }

};

int main(){

    StuList a;

    SubList b;

    for(;;){

        cout<<"Enter : \n1. To add subject\n2. To add student\n3. To assign subject to a particular student\n4. To show all students for a particular subject\n5. To show all subjects of a particular student\n6. Exit\n";

        int c;

        cin>>c;

        switch(c){

            case 1:{

                cout<<"Enter subject code and name\n";

                int x;

                string y;

                cin>>x>>y;

                b.push(x,y);

                break;

            }

            case 2:{

                cout<<"Enter student roll, pHNo and name\n";

                int x,z;

                string y;

                cin>>x>>z>>y;

                a.push(x,y,z);

                break;

            }

            case 3:{

                cout<<"Available subjects: \n";

                b.display();

                cout<<"Enter student roll and subject code \n";

                int x,y;

                cin>>x>>y;

                a.add(x,y);

                b.add(y,x);

                break;

            }

            case 4:{

                cout<<"Enter subject code \n";

                int x;

                cin>>x;

                vector<int> l(b.disStu(x));

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

                for (auto i = l.begin(); i != l.end(); ++i)

                a.display(\*i);

                break;

            }

            case 5:{

                cout<<"Enter student roll \n";

                int x;

                cin>>x;

                vector<int> p(a.disSub(x));

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

                for (auto j = p.begin(); j != p.end(); ++j)

                b.display(\*j);

                break;

            }

            case 6:{

                cout<<"Exiting ...";

                exit(0);

                break;

            }

            default:{

                cout<<"Wring Input";

                break;

            }

        }

    }

    return 0;

}

Output:

