# Group Details :-

K.A.S.Sekhar 1601-14-733-091

***GROUP 7***

Dasaradh Ram Reddy 1601-14-733-092 Eshwar 1601-14-733-093

Harsha Vardhan 1601-14-733-094

G.S.Harsha Anirudh 1601-14-733-095

# TEAM LEADER:- Mr. Sekhar Karedla 1601-14-733-091

**Program 1:-**

//a program to implement the matrix operations.

/\*Solution to this program is developed by BE ¼ CSE-2 Group-7(Sekhar, Dasaradh, Eshwar, Harsha and Harshavardhan)\*/

#include<iostream>

#include<stdlib.h> using namespace std; class matrix

{

int m,n,p,q,\*\*a,\*\*b,m1,n1,\*\*c; // using pointer to pointers for 2d array public:

void getdata(); void display(); void add(); void subtract(); void multiply();

};

void matrix::getdata()

{

cout<<"enter the order of the first matrix : "; cin>>m>>n;

cout<<"enter the order of the second matrix : "; cin>>p>>q;

a = new int \*[m]; //allocating memory for 2-D array dynamically

for(int i = 0; i < m; ++i) a[i] = new int[n];

b = new int\*[p]; for(int i = 0; i < p; ++i) b[i] = new int[q];

int i,j;

cout<<"enter the elements of the first matrix : \n"; for(i=0;i<m;i++)

for(j=0;j<n;j++) cin>>a[i][j];

cout<<"enter the values of the second matrix : \n"; for(i=0;i<p;i++)

for(j=0;j<q;j++) cin>>b[i][j];

}

void matrix::display()

{

int i,j;

cout<<"the matrix is \n"; for(i=0;i<m1;i++)

{

cout<<endl; for(j=0;j<n1;j++) cout<<c[i][j]<<" ";

}

}

void matrix::add()

{

if(m!=p||n!=q)

{

cout<<"wrong order : "; exit(-1);

}

int i,j; m1=m;n1=n;

c = new int\*[m1];

for(int i = 0; i < m1; ++i) c[i] = new int[n1]; for(i=0;i<m1;i++) for(j=0;j<n1;j++) c[i][j]=a[i][j]+b[i][j];

}

void matrix::subtract()

{

if(m!=p||n!=q) //exceptions

{

cout<<"wrong order : "; exit(-1);

}

int i,j; m1=m;n1=n;

c = new int\*[m1];

for(int i = 0; i < m1; ++i) c[i] = new int[n1]; for(i=0;i<m1;i++) for(j=0;j<n1;j++) c[i][j]=a[i][j]-b[i][j];

}

void matrix::multiply()

{

if(n!=p) //exception for multiplication

{

cout<<"wrong order : "; exit(-1);

}

int i,j,k; m1=m;n1=q;

c = new int\*[m1];

for(int i = 0; i < m1; ++i) c[i] = new int[n1]; for(i=0;i<m1;i++)

{

for(j=0;j<n1;j++)

{

c[i][j]=0; for(k=0;k<n;k++) c[i][j]+=a[i][k]\*b[k][j];

}

}

}

int main()

{

matrix s; s.getdata(); int ch;

cout<<"press 1 for addition , 2 for subtraction ,3 for multiplication , 4 for division ";

cin>>ch;

switch(ch) //using the switch case for menu purpose

{

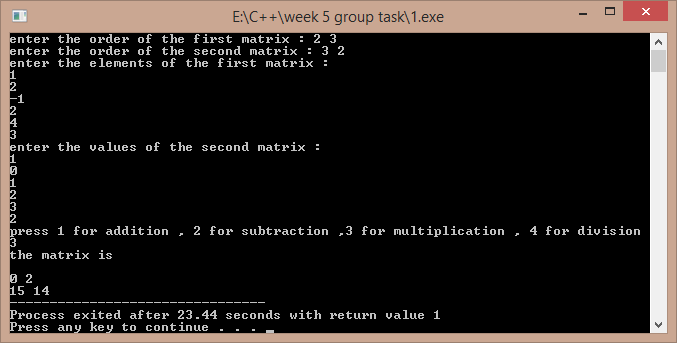
}

return 1;

}

case 1: s.add();s.display();break;

case 2: s.subtract();s.display();break; case 3:s.multiply();s.display();break; default:cout<<"wrong choice ...";



# Program 2:-

**//**a program to implement operations on complex numbers

/\*Solution to this program is developed by BE ¼ CSE-2 Group-7(Sekhar, Dasaradh, Eshwar, Harsha and Harshavardhan)\*/

#include<iostream> using namespace std; class complex

{

private:

float real; float imag;

public:

void add(complex,complex); void sub(complex,complex); void mul(complex,complex); void divide(complex,complex); void display();

void getdata(float,float);

};

void complex::add(complex x, complex y)

{

this->real=x.real+y.real; //using this pointer to point towards its own member this->imag=x.imag+y.imag;

}

void complex::sub(complex x, complex y)

{

this->real=x.real-y.real; this->imag=x.imag-y.imag;

}

void complex::mul(complex x, complex y)

{

this->real=(x.real\*y.real)-(x.imag)\*(y.imag); this->imag=(x.real\*y.imag)+(y.real\*x.imag);

}

void complex::divide(complex x, complex y)

{

this-

>real=((x.real\*y.real)+(x.imag)\*(y.imag))/((y.real\*y.real)+(y.imag\*y.imag)); this->imag=((y.real\*x.imag)-

(x.real\*y.imag))/((y.real\*y.real)+(y.imag\*y.imag));

}

void complex::display()

{

if(imag>0) cout<<real<<"+"<<imag<<"i\n"; else

cout<<real<<"-"<<imag<<"i\n";

}

void complex::getdata(float r,float i)

{

real=r; imag=i;

}

int main()

{

complex c1,c2,c3; //two complex numbers for storing the data float m,n; //c3 for storing the result

cout<<"\nenter the real and imaginery part of first complex no. : "; cin>>m>>n;

c1.getdata(m,n);

cout<<"\nenter the real and imaginery part of second complex no. : "; cin>>m>>n;

c2.getdata(m,n); int ch;

cout<<"\nenter 1 for addition\nenter 2 for subtraction\nenter 3 for

subtraction\nenter 4 division :\n"; cin>>ch;

switch(ch) //switch case for the menu

{

case 1:

{

c3.add(c1,c2);

}break; case 2:

{

c3.sub(c1,c2);

}break; case 3:

{

c3.mul(c1,c2);

}break; case 4:

{

c3.divide(c1,c2);

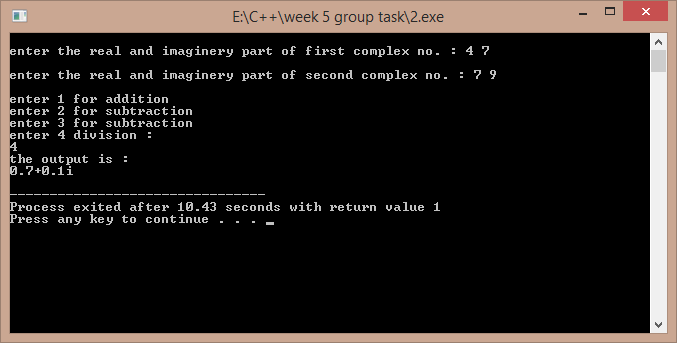
}break; default:cout<<"wrong choice ....";

}

cout<<"the output is : \n"; c3.display();

return 1;

}



# Program 3:-

//a sample program to show the working of a bank program

/\*Solution to this program is developed by BE ¼ CSE-2 Group-7(Sekhar, Dasaradh, Eshwar, Harsha and Harshavardhan)\*/

#include<iostream>

#include<string.h>

#include<stdlib.h> using namespace std; class bank

{

long int ac; float bal;

char t,name[100],pass[100]; public:

void withdraw(float); void deposite(float); void display();

void create();

friend void assign(bank []); //two friend functions to access the friend int check(bank [],char [],long int); // private data of objects

};

void bank::display() // a function to display the record

{

cout<<"\n\n\nName : "<<name; cout<<"\nAccount no : "<<ac;

cout<<"\nAvailable Balance : "<<bal; cout<<"\nAccount Type : "<<t;

}

void bank::create() //a function to create a new account

{

cout<<"\nenter you name : "; cin>>name;

cout<<"\nenter your account number : "; cin>>ac;

cout<<"\nenter the balance : "; cin>>bal;

cout<<"\ncreate a new password : "; cin>>pass;

while(bal<1000)

{

amount ...";

}

cout<<"\nminimum amount should be 1000 , please re- enter cin>>bal;

cout<<"\nenter the type of account : "; cin>>t;

}

void bank::withdraw(float w)

{

if(w>bal)

{

cout<<"\nnot enough balance..."; //exception if there is not enough exit(-1); //balance

}

cout<<"\npresent balance is : "<<bal; cout<<"\nwithdrawal amount : "<<w; cout<<"\nnew balance is : "<<bal-w; bal=bal-w;

}

void bank::deposite(float w)

{

cout<<"\npresent balance : "<<bal; cout<<"\ndeposite amount : "<<w; cout<<"\nnew amount : "<<bal+w; bal=bal+w;

}

void assign(bank b[]) //a function to assign some existing accounts to objects

{

strcpy(b[0].name,"sekhar"); b[0].ac=23095; b[0].bal=9023099.4326; b[0].t='s';

strcpy(b[0].pass,"karedla1");

strcpy(b[1].name,"ram"); b[1].ac=23021; b[1].bal=90099.4326; b[1].t='s';

strcpy(b[1].pass,"shayam2");

strcpy(b[2].name,"shyam"); b[2].ac=23675; b[2].bal=10299.426; b[2].t='c';

strcpy(b[2].pass,"ram3");

}

int check(bank b[],char p[],long int z)

{ //a friend function to check acc no . and its corresponding password int i;

for(i=0;i<3;i++)

{

if(strcmp(b[i].pass,p)==0&&(b[i].ac==z)) return i;

}

return -1;

}

int main()

{

bank b[3]; assign(b); int ch;

cout<<"\nenter 1 to create an account\nenter 2 to deposite amount\nenter 3 to withdraw amount : ";

cin>>ch;

switch(ch) //a menu to create , with draw , create new account

{

case 1:

{

bank s; s.create();

}break; case 2:

{

char p[100];int k;long int z; cout<<"\nenter your account number : "; cin>>z;

cout<<"\nenter your pass-word : "; cin>>p;

k=check(b,p,z); if(k>=0)

{

}

else

{

b[k].display();

cout<<"no matching profile ..."; exit(-1);

}

cout<<"\nenter the amount to be deposited : "; float w;

cin>>w; b[k].deposite(w);

}break; case 3:

{

char p[100];int k;long int z; cout<<"\nenter your account number : "; cin>>z;

cout<<"\nenter your pass-word : "; cin>>p;

k=check(b,p,z); if(k>=0)

{

}

else

{

b[k].display();

cout<<"no matching profile ..."; exit(-1);

}

cout<<"\nenter the amount to be withdrawn : "; float w;

cin>>w; b[k].withdraw(w);

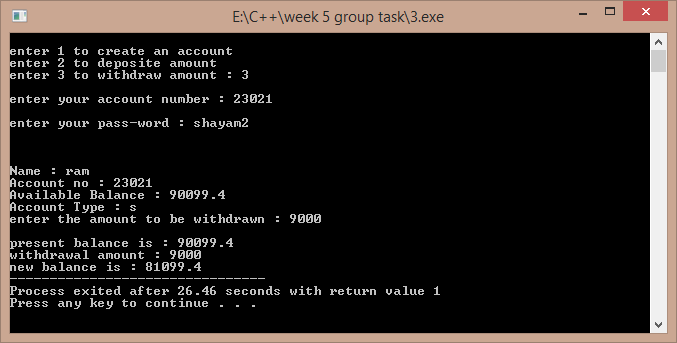
}break;

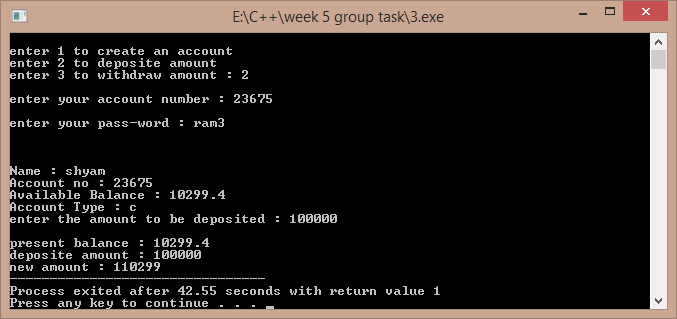
default : cout<<"wrong option entry ....";

}

return 1;

}





Another sample:-

# Program 4:-

//a sample program to demonstrate operations on vectors

/\*Solution to this program is developed by BE ¼ CSE-2 Group-7(Sekhar, Dasaradh, Eshwar, Harsha and Harshavardhan)\*/

#include<iostream> using namespace std;

class vector1

{

int \*v,n; public:

void create(); void modify(); void multiply(); void display();

};

void vector1::create() //creating a vector of required size

{

int i;

cout<<"enter the size of the vector : "; cin>>n;

v=new int[n];

cout<<"\nenter the elements of the vector : \n"; for(i=0;i<n;i++)

cin>>v[i];

}

void vector1::modify() // modifying its elements

{

int p,ch;

cout<<"\nenter the element to be modified : "; cin>>p;

if(p>n) //the program terminates if this exception occurs

{

cout<<"element number out of bounds ..."; exit(-1);

}

cout<<"\nenter the modification : "; cin>>ch;

v[p-1]=ch;

}

void vector1::multiply() // multiplying its elements with a scalar

{

int s,i;

cout<<"\nenter the scalar to multiplied to the whole vector : "; cin>>s;

for(i=0;i<n;i++) v[i]=v[i]\*s;

}

void vector1::display() // display the vector

{

int i;

cout<<"\n the vector is : \n"; for(i=0;i<n;i++)

cout<<" "<<v[i];

}

int main()

{

vector1 m; m.create(); int flag=0; do

{

cout<<"\nenter 1 so as to modify an element \nenter 2 so as to multiply the vector by a scalar\nenter 3 so as to display the vector\nenter 4 to quit\n";

int ch; cin>>ch; switch(ch)

{

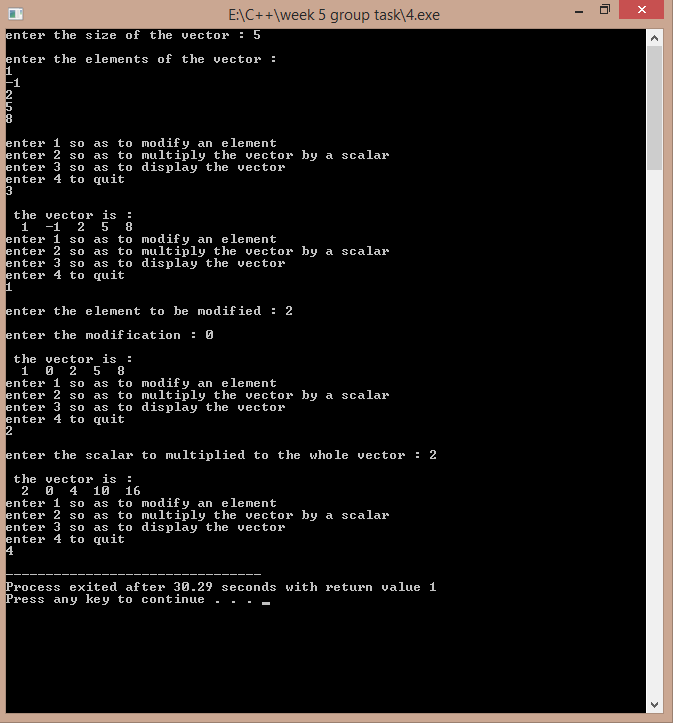
case 1:m.modify();m.display();break; case 2:m.multiply();m.display();break; case 3:m.display();break;

default : flag=1;

}

}while(flag==0); return 1;

}



# Program 6:-

//a program to implement the displaying the marks of the students

/\*Solution to this program is developed by BE ¼ CSE-2 Group-7(Sekhar, Dasaradh, Eshwar, Harsha and Harshavardhan)\*/

#include<iostream> using namespace std; int m1,m2;

class student

{

char name[80],add[100]; int rno,t;

float avg;

int m[6]; public:

student()

{

avg=0;t=0;

}

void readata()

{

int i;

cout<<"\nenter name : "; cin>>name; cout<<"\nenter roll no : "; cin>>rno;

cout<<"\nenter the address : "; cin>>add;

cout<<"\nenter the marks of 6 subjects : "; for(i=0;i<6;i++)

cin>>m[i];

}

void percent()

{

int i; for(i=0;i<6;i++) t+=m[i]; avg=t/6.0;

}

void display()

{

int i;

cout<<"\n\n\nNAME : "<<name; cout<<"\nADDRESS : "<<add; cout<<"\nROLL NO : "<<rno; cout<<"\nTOTAL MARKS : "<<t;

cout<<"\nPERCENTAGE : "<<avg; cout<<"\n marks of individual subjects : \n"; for(i=0;i<6;i++)

{

cout<<"\nmarks subject "<<i+1<<" : "<<m[i];

}

}

friend void class\_average(student s[],int n)

{m1=0;float cavg,t; // a friend function to to calculate class for(int i=0;i<n;i++) // highest and lowest

{

if(m1<s[i].avg)

m1=s[i].avg;

}

m2=m1;

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

{

}t=0;

if(m2>s[i].avg)

m2=s[i].avg;

for(int i=0;i<n;i++) t+=s[i].avg; cavg=float(t/n);

cout<<"\nthe minimum is : "<<m2; cout<<"\nthe maximum is : "<<m1; cout<<"\nclass average is : "<<cavg;

}

friend void decending(student s[],int n)

{

float \*avg1=new float[n];

for(int i=0;i<n;i++) //to display the student avg1[i]=s[i].avg; //record in descending order int i,j,k1=0,f=0;

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

{m1=0;

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

{

if(m1<s[i].avg)

{

k1=i; m1=s[i].avg;

}

}

s[k1].display(); s[k1].avg=0;

}

for(int i=0;i<n;i++) s[i].avg=avg1[i];

/\* student temp; int i,j; for(i=0;i<n;i++)

{

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

{

if(s[j].avg<s[j+1].avg)

{

}

}\*/

temp=s[j]; s[j]=s[j+1]; s[j+1]=temp;

}

}

friend void topn(student s[],int n,int p)

{

}

}

};

int main()

{

float \*avg1=new float[n];

for(int i=0;i<n;i++) //a function to display the top n avg1[i]=s[i].avg; //students only

int i,j,k1=0,f=0; for(j=0;j<n;j++)

{m1=0;

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

{

if(m1<s[i].avg)

{

k1=i; m1=s[i].avg;

}

}

if(f<=p)

{

s[k1].display();

s[k1].avg=0;f++;

}

else break;

for(int i=0;i<n;i++) s[i].avg=avg1[i];

int n;

cout<<"\nenter the number of students : "; cin>>n;

student \*s=new student[n];

cout<<"\nenter the data of "<<n<<" students : "; int i;

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

{

}

int ch;

s[i].readata();

s[i].percent();

cout<<"\nenter 1 display all student details \nenter 2 descending order view \nenter 3 to see top n students \nenter 4 to display class average , lowest

,highest : \n";

cin>>ch; switch(ch)

{

case 1:

{

for(i=0;i<n;i++) s[i].display();

}break; case 2:

{

decending(s,n);

}break; case 3:

{

int p;

cout<<"\nenter the value of n : "; cin>>p;

topn(s,n,p);

}break; case 4:

{

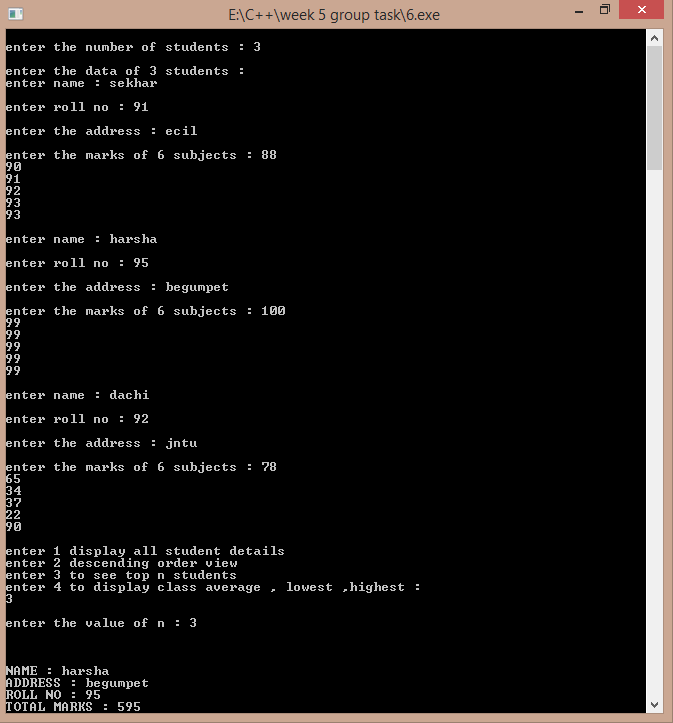
}

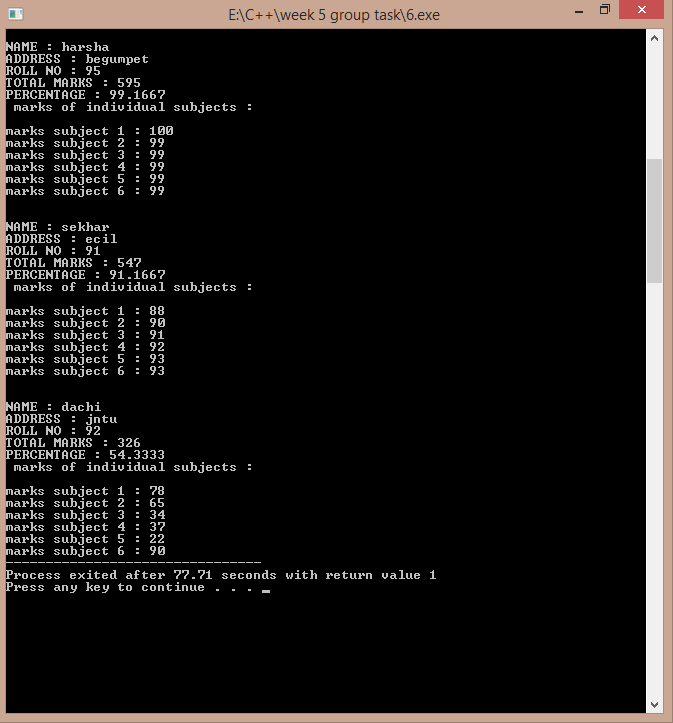
return 1;

}

class\_average(s,n);

}break; default:cout<<"wrong choice .....";





Another sample:-

