**Chapter-06**

**Problem No.-01**

**Code:**

#include<iostream>

#include<iomanip>

using namespace std;

class Int

{

private:

int V;

public:

Int():V(0)

{ }

Int(int v):V(v)

{ }

void display();

void add(Int,Int);

};

void Int::display()

{

cout<<"Value: "<<V<<endl;

}

void Int::add(Int v1,Int v2)

{

V=v1.V+v2.V;

}

int main()

{

Int num1,num2(7),num3(17);

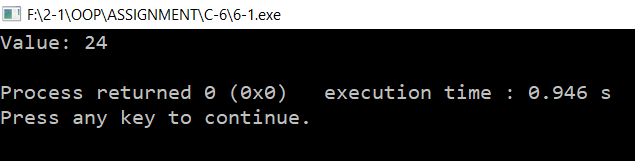
num1.add(num2, num3);

num1.display();

return 0;

}

**Output:**

****

**Problem No.-03**

**Code:**

#include<iostream>

using namespace std;

class time

{

private:

int hours,minutes,seconds;

public:

time():hours(0), minutes(0), seconds(0)

{ }

time(int h, int m, int s):hours(h), minutes(m), seconds(s)

{ }

void display() const;

void add(time, time);

};

void time::display() const

{

cout <<"Time: "<<hours <<":" <<minutes <<":" <<seconds<<endl;

}

void time::add(time t1, time t2)

{

seconds = t1.seconds + t2.seconds;

if(seconds > 59)

{

seconds -= 60;

minutes++;

}

minutes += t1.minutes + t2. minutes;

if(minutes > 59)

{

minutes -= 60;

hours++;

}

hours += t1.hours + t2.hours;

}

int main()

{

const time tm1(9, 05, 45);

const time tm2(5, 52, 17);

time tm3;

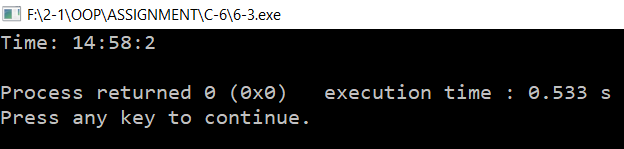
tm3.add(tm1, tm2);

tm3.display();

return 0;

}

**Output:**

****

**Problem No.-05**

**Code:**

#include <iostream>

#include<iomanip>

using namespace std;

class date

{

private:

int month,day,year;

public:

void get();

void display();

};

void date::get()

{

cout <<"Enter date in mm/dd/yy format: ";

cin >>month >>day >>year;

}

void date::display()

{

cout <<"Date: " <<month <<"/" <<day <<"/" <<year<<endl;

}

int main()

{

date d;

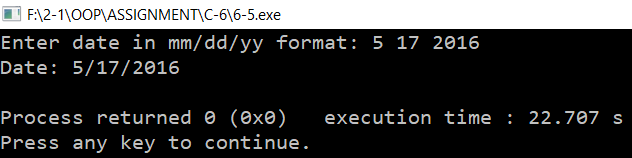
d.get();

d.display();

return 0;

}

**Output:**

****

**Problem No.-07**

**Code:**

#include<iostream>

#include<iomanip>

using namespace std;

class angle

{

private:

int degrees;

float minutes;

char direction;

public:

angle():degrees(0),minutes(0),direction(0)

{ }

angle(int deg,float m,char d):degrees(deg),minutes(m),direction(0)

{ }

void getdata();

void display();

};

void angle::getdata()

{

cout <<"Enter degrees: ";

cin >>degrees;

cout <<"Enter minutes: ";

cin >>minutes;

cout <<"Enter directions(E, W, N, S): ";

cin >>direction;

}

void angle::display()

{

cout <<degrees <<"\xF8" <<minutes <<"\' " <<direction;

if(direction == 'E' || direction == 'e' || direction == 'W' || direction == 'w')

cout <<" Longitude !";

if(direction == 'N' || direction == 'n' || direction == 'S' || direction == 's')

cout <<" Latitude !";

}

int main()

{

angle c1;

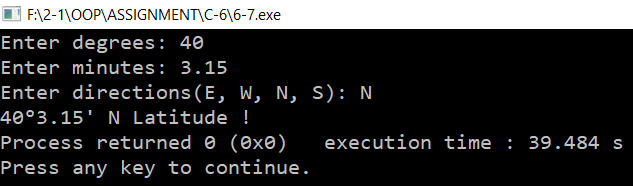
c1.getdata();

c1.display();

return 0;

}

**Output:**

****

**Problem No.-09**

**Code:**

#include<iostream>

#include<iomanip>

using namespace std;

class fraction

{

private:

int numerator,denominator;

char ch;

public:

fraction():numerator(0),denominator(0),ch(0)

{ }

fraction(int num, int denom, char c):numerator(num),denominator(denom),ch(c)

{ }

void get();

void add(fraction, fraction);

void display();

};

void fraction::get()

{

cout<<"Enter fraction: ";

cin>>numerator>>ch>>denominator;

}

void fraction::add(fraction f1,fraction f2)

{

numerator=f1.numerator\*f2.denominator+f2.numerator\*f1.denominator;

denominator=f1.denominator\*f2.denominator;

}

void fraction::display()

{

cout<<"Addition: "<<numerator<<"/"<<denominator<<endl;

}

int main()

{

char t;

fraction frac1,frac2,frac3;

do

{

frac1.get();

frac2.get();

frac3.add(frac1,frac2);

frac3.display();

cout <<"\n\nDo you want to continue(y,n): ";

cin >>t;

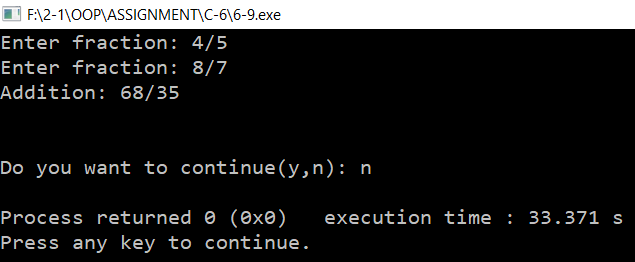
}

while(t == 'Y' || t== 'y');

return 0;

}

**Output:**

****

**Problem No-11**

**Code:**

#include<iostream>

#include<cmath>

using namespace std;

class fraction

{

private:

int num, den;

public:

fraction():num(0),den(0)

{ }

fraction(int n, int d):num(n), den(d)

{ }

void get\_frac();

void lowterms();

void add\_frac(fraction,fraction);

void sub\_frac(fraction,fraction);

void mul\_frac(fraction,fraction);

void div\_frac(fraction,fraction);

};

void fraction::get\_frac()

{

char ch;

cin >> num >> ch >> den;

}

void fraction::lowterms()

{

long tnum, tden, temp, gcd;

tnum = labs(num);

tden = labs(den);

if (tden == 0)

{

cout << "Illegal fraction : division by 0";

exit(1);

}

else if (tnum == 0)

{

num = 0;

den = 1;

return;

}

while (tnum != 0)

{

if (tnum < tden)

{

temp = tnum;

tnum = tden;

tden = temp;

}

tnum = tnum - tden;

}

gcd = tden;

num = num / gcd;

den = den / gcd;

}

void fraction::add\_frac(fraction a,fraction b)

{

num = a.num\*b.den + a.den\*b.num;

den = a.den\*b.den;

lowterms();

cout << a.num << " / " << a.den<<" + "<<b.num<<" / "<<b.den<<" = "<<num<<" / "<<den<<endl;

}

void fraction::sub\_frac(fraction a,fraction b)

{

num = a.num\*b.den - a.den\*b.num;

den = a.den\*b.den;

lowterms();

cout << a.num << " / " << a.den<<" - "<<b.num<<" / "<<b.den<<" = "<<num<<" / "<<den<<endl;

}

void fraction::mul\_frac(fraction a,fraction b)

{

num = a.num\*b.num;

den = a.den\*b.den;

lowterms();

cout << a.num << " / " << a.den<<" \* "<<b.num<<" / "<<b.den<<" = "<<num<<" / "<<den<<endl;

}

void fraction::div\_frac(fraction a, fraction b)

{

num = a.num\*b.den;

den = a.den\*b.num;

lowterms();

cout << a.num << " / " << a.den<<" / "<<b.num<<" / "<<b.den<<" = "<<num<<" / "<<den<<endl;

}

int main()

{

fraction a, b, c,d,e,f;

char t;

cout << "Enter fraction A: ";

a.get\_frac();

cout<<"Enter an operator(+, -, \*, /): ";

cin>>t;

cout << "Enter fraction B: ";

b.get\_frac();

switch(t)

{

case '+':

c.add\_frac(a, b);

break;

case '-':

d.sub\_frac(a, b);

break;

case '\*':

e.mul\_frac(a, b);

break;

case '/':

f.div\_frac(a, b);

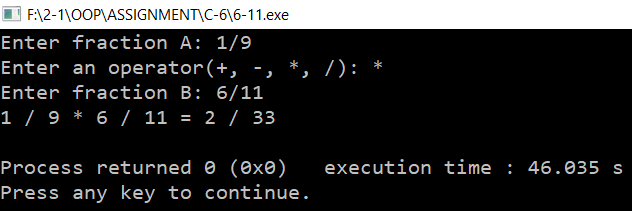
break;

}

return 0;

}

**Output:**

****

**Chapter-07**

**Problem No.-01**

**Code:**

#include<iostream>

#include<string>

#include<string.h>

using namespace std;

void reversit(char \*);

int main()

{

char ara[100];

cout<<"Enter a string to reverse: ";

gets(ara);

reversit(ara);

cout<<"Reverse: "<<ara<<endl;

return 0;

}

void reversit(char \*str)

{

char temp;

int i,len;

len=strlen(str)-1;

for(i=0; i<len; i++,len--)

{

temp=str[i];

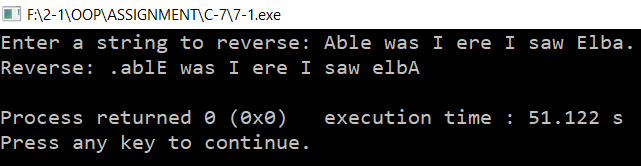
str[i]=str[len];

str[len]=temp;

}

}

**Output:**

****

**Problem No.-03**

**Code:**

#include <iostream>

using namespace std;

class Distance

{

private:

int feet;

float inches;

public:

Distance():feet(0),inches(0)

{ }

Distance(int f, float in):feet(f),inches(in)

{ }

void getdist()

{

cout << "\n Enter feet: ";

cin >> feet;

cout << " Enter inches: ";

cin >> inches;

}

void showdist() const

{

cout << feet << "\'-" << inches << "\" ";

}

void add\_dist(Distance,Distance);

void div\_dist(Distance, int);

};

void Distance::add\_dist(Distance d2, Distance d3)

{

inches = d2.inches + d3.inches;

feet = 0;

if(inches >= 12.0)

{

inches -= 12.0;

feet++;

}

feet += d2.feet + d3.feet;

}

void Distance::div\_dist(Distance d2, int divisor)

{

float fltfeet = d2.feet + d2.inches/12.0;

fltfeet /= divisor;

feet = int(fltfeet);

inches = (fltfeet-feet) \* 12.0;

}

int main()

{

Distance dist[100]; Distance total(0, 0.0), average;

int n=0;

char ans;

cout << endl;

do

{

cout << "Enter distance number " << n+1;

dist[n++].getdist();

cout << "Enter another (y/n)?: ";

cin >> ans;

}

while( ans != 'n' );

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

{

total.add\_dist(total,dist[j]);

}

total.showdist();

average.div\_dist(total,n);

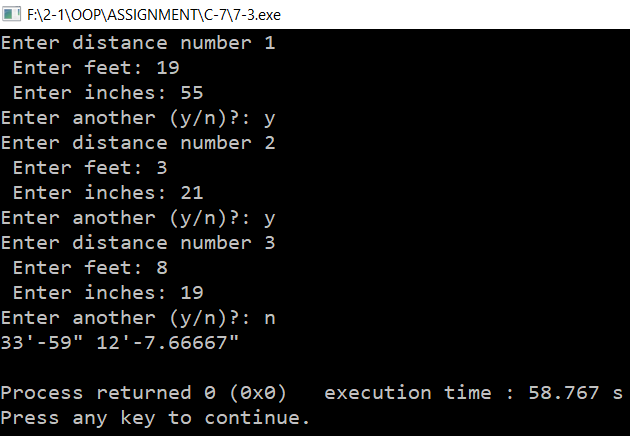
average.showdist();

cout << endl;

return 0;

}

**Output:**

****

**Problem No.-05**

**Code:**

#include<iostream>

#include<cmath>

using namespace std;

class fraction

{

private:

int num, den;

public:

fraction():num(0),den(0)

{ }

fraction(int n, int d):num(n), den(d)

{ }

void get\_frac();

void lowterms();

void disp();

void add\_frac(fraction,fraction);

void div\_frac(fraction,fraction);

};

void fraction::get\_frac()

{

char ch; cin >> num >> ch >> den;

}

void fraction::disp()

{

cout<<num<<" / "<<den<<endl;

}

void fraction::lowterms()

{

long tnum, tden, temp, gcd;

tnum = labs(num);

tden = labs(den);

if (tden == 0)

{

cout << "Illegal fraction : division by 0"; exit(1);

}

else if (tnum == 0)

{

num = 0; den = 1;

return;

}

while (tnum != 0)

{

if (tnum < tden)

{

temp = tnum;

tnum = tden;

tden = temp;

}

tnum = tnum - tden;

}

gcd = tden;

num = num / gcd;

den = den / gcd;

}

void fraction::add\_frac(fraction a,fraction b)

{

num = a.num\*b.den + a.den\*b.num;

den = a.den\*b.den;

lowterms();

cout << a.num << " / " << a.den<<" + "<<b.num<<" / "<<b.den<<" = "<<num<<" / "<<den<<endl;

}

void fraction::div\_frac(fraction a, fraction b)

{

num = a.num\*b.den;

den = a.den\*b.num;

lowterms();

cout << a.num << " / " << a.den<<" / "<<b.num<<" / "<<b.den<<" = "<<num<<" / "<<den<<endl;

}

int main()

{

fraction ara[100],sum,divisor,result;

char ch;

int i=0;

do

{

cout << "Enter a fraction: ";

ara[i].get\_frac();

i++;

cout << "Enter again?(y/n):"; cin>>ch;

}

while (ch == 'y'&& i < 100);

sum = { 0, 1 };

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

sum.add\_frac(sum, ara[j]);

divisor = { 1, i + 1 };

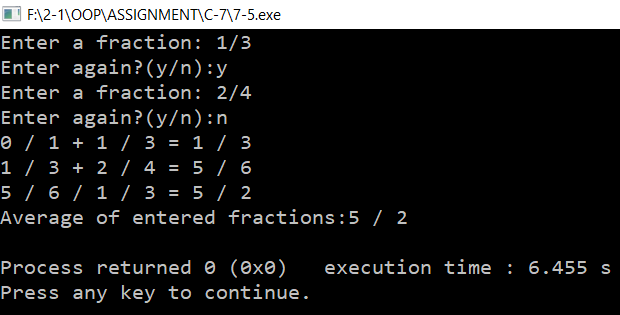
result.div\_frac(sum, divisor);

cout << "Average of entered fractions:";

result.disp();

}

**Output:**

****

**Problem No.-07**

**Code:**

#include<iostream>

#include<conio.h>

#include<stdlib.h>

#include<string>

#include<string.h>

using namespace std;

long mstold(char str[])

{

string s=" 0123456789";

char str1[20];

for(int i=0, j=0; i<strlen(str); i++)

if(s.find(str[i])<20)

str1[j++]=str[i];

return atol(str1);

}

int main()

{

char ara[20];

do

{

cout<<"Enter a monetary value: "; cin>>ara;

cout<<mstold(ara)<<endl;

cout<<"Press y to continue"<<endl;

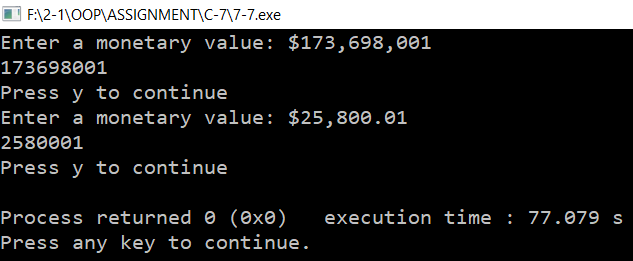
}

while(getch()=='y');

return 0;

}

**Output:**



**Problem No.-09**

**Code:**

#include <iostream>

using namespace std;

class Queue

{

private:

enum { MAX = 10 };

int st[MAX];

int fro,rear;

public:

Queue()

{

fro=rear=-1;

}

void put(int var)

{

if(rear==MAX-1)

{

cout<<"Queue overflow"<<endl;

}

else if(fro==-1 && rear==-1)

{

fro=rear=0;

st[rear]=var;

cout<<"Item inserted: "<<var<<endl;

}

else

{

rear++;

st[rear]=var;

cout<<"Item inserted: "<<var<<endl;

}

}

void get()

{

if(fro==-1)

{

cout<<"Queue is empty"<<endl;

}

else

{

cout<<"Queue items: "<<endl;

for(int i=fro; i<=rear; i++)

{

cout<<st[i]<<" ";

}

cout<<endl;

}

}

};

int main()

{

Queue s1;

s1.put(11);

s1.put(22);

s1.get();

s1.put(33);

s1.put(44);

s1.put(55);

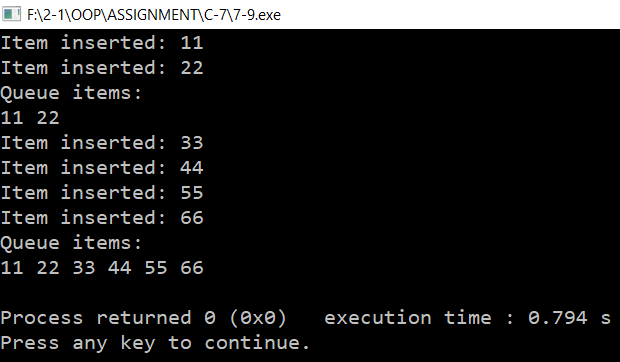
s1.put(66);

s1.get();

return 0;

}

**Output:**

****

**Problem No.-11**

**Code:**

#include <iostream>

#include <string>

#include <sstream>

#include <iomanip>

using namespace std;

string ldtoms(long double sum)

{

stringstream ss (stringstream::in | stringstream::out);

ss.setf(ios::fixed);

ss << setprecision(2) << sum;

string s = ss.str();

s.insert(0, "$");

int p = s.find('.') - 1, n = 0;

for(int i = p; i > 0; i--)

if(++n % 3 == 0)

s.insert(i, ",");

return s;

}

int main()

{

long double n;

char ch;

do

{

cout<<"Enter amount: "; cin>>n;

string ara= ldtoms(n);

cout<<"Money: "<<ara<<endl;

cout<<"Continue? (y/n): "; cin>>ch;

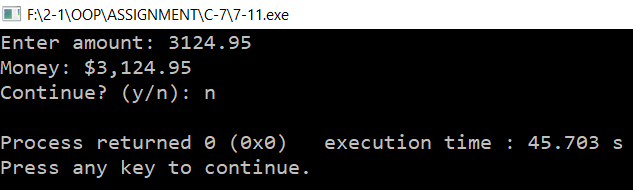
}

while(ch!='n');

return 0;

}

**Output:**

****

**Chapter-08**

**Problem No.-01**

**Code:**

#include <iostream>

using namespace std;

class Distance

{

private:

int feet;

float inches;

public:

Distance() : feet(0), inches(0.0)

{ }

Distance(int ft, float in) : feet(ft), inches(in)

{ }

void getdist()

{

cout << "Enter feet: "; cin >> feet;

cout << "Enter inches: "; cin >> inches;

}

void showdist() const

{

cout << feet << "\'-" << inches << "\" ";

}

Distance operator - ( Distance ) const;

};

Distance Distance::operator - (Distance d2) const

{

int f = feet - d2.feet;

float i = inches - d2.inches;

if(i<0)

{

i += 12.0; f--;

}

return Distance(f,i);

}

int main()

{

Distance dist1, dist3; dist1.getdist();

Distance dist2(11, 6.25);

dist3 = dist1 - dist2;

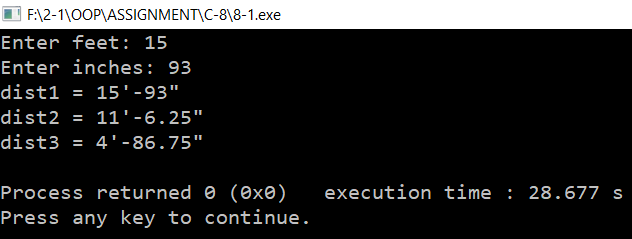
cout << "dist1 = "; dist1.showdist(); cout << endl;

cout << "dist2 = "; dist2.showdist(); cout << endl;

cout << "dist3 = "; dist3.showdist(); cout << endl;

}

**Output:**

****

**Problem No.-03**

**Code:**

#include<iostream>

using namespace std;

class time

{

private:

int hours,minutes,seconds;

public:

time():hours(0), minutes(0), seconds(0)

{ }

time(int h, int m, int s):hours(h), minutes(m), seconds(s)

{ }

void display() const;

time operator + (time) const;

};

void time::display() const

{

cout <<"Time: "<<hours <<":" <<minutes <<":" <<seconds<<endl;

}

time time::operator + (time t1) const

{

int ts=seconds+t1.seconds;

int tm=minutes+t1.minutes;

int th=hours+t1.hours;

if(ts>59)

{

ts-=60;

tm++;

}

if(tm>59)

{

tm-=60;

th++;

}

return time(th,tm,ts);

}

int main()

{

const time tm1(9, 9, 45);

const time tm2(3, 52, 17);

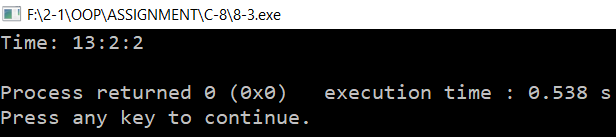
time tm3=tm1+tm2;

tm3.display();

return 0;

}

**Output:**

****

**Problem No.-05**

**Code:**

#include<iostream>

using namespace std;

class time

{

private:

int hours,minutes,seconds;

public:

time():hours(0), minutes(0), seconds(0)

{ }

time(int h, int m, int s):hours(h), minutes(m), seconds(s)

{ }

void display();

time operator + (time);

time operator ++ () ;

time operator ++ (int);

time operator -- ();

time operator -- (int);

};

time time::operator ++ ()

{

++seconds;

if(seconds>59)

{

++minutes;

seconds-=60;

}

if(minutes>59)

{

++hours;

minutes-=60;

}

return time(hours,minutes,seconds);

}

time time::operator ++ (int )

{

time T(hours,minutes,seconds);

++seconds;

if(seconds>59)

{

++minutes;

seconds-=60;

}

if(minutes>59)

{

++hours;

minutes-=60;

}

return T;

}

time time::operator -- ()

{

--seconds;

if(seconds<0)

{

--minutes;

seconds+=60;

}

if(minutes<0)

{

--hours;

minutes+=60;

}

return time(hours,minutes,seconds);

}

time time::operator -- (int )

{

time T(hours,minutes,seconds);

--seconds;

if(seconds<0)

{

--minutes;

seconds+=60;

}

if(minutes<0)

{

--hours;

minutes+=60;

}

return T;

}

void time::display()

{

cout <<"Time: "<<hours <<":" <<minutes <<":" <<seconds<<endl;

}

time time::operator + (time t1)

{

int ts=seconds+t1.seconds;

int tm=minutes+t1.minutes;

int th=hours+t1.hours;

if(ts>59)

{

ts-=60;

tm++;

}

if(tm>59)

{

tm-=60;

th++;

}

return time(th,tm,ts);

}

int main()

{

time tm1(9, 7, 45);

time tm2(3, 52, 14);

time tm3=tm1+tm2;

tm3.display();

cout<<"After prefix Increment: "<<endl;

time tm4=++tm3;

tm3.display();

tm4.display();

cout<<"After postfix Increment: "<<endl;

time tm5=tm3++;

tm3.display();

tm5.display();

cout<<"After prefix Decrement: "<<endl;

time tm6=--tm3;

tm3.display();

tm6.display();

cout<<"After postfix Decrement: "<<endl;

time tm7=tm3--;

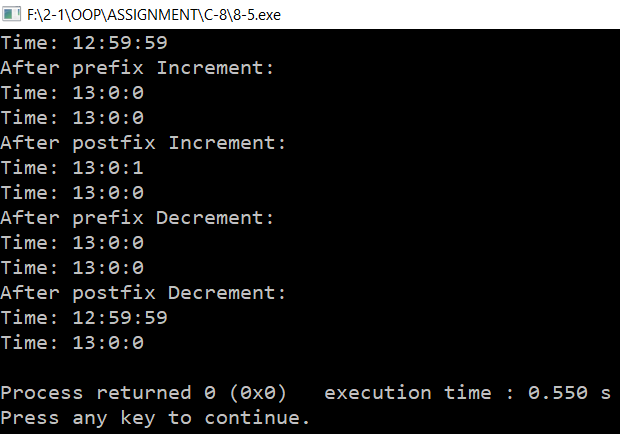
tm3.display();

tm7.display();

return 0;

}

**Output:**

****

**Problem No.-07**

**Code:**

#include<iostream>

#include<cmath>

using namespace std;

class fraction

{

private:

int num, den;

public:

fraction():num(0),den(0)

{ }

fraction(int n, int d):num(n), den(d)

{ }

void get\_frac();

void lowterms();

void display();

fraction operator + (fraction);

fraction operator - (fraction);

fraction operator \* (fraction);

fraction operator / (fraction);

};

void fraction::get\_frac()

{

char ch; cin >> num >> ch >> den;

}

void fraction::display()

{

cout<<num<<" / "<<den;

}

void fraction::lowterms()

{

long tnum, tden, temp, gcd;

tnum = labs(num);

tden = labs(den);

if (tden == 0)

{

cout << "Illegal fraction : division by 0";

exit(1);

}

else if (tnum == 0)

{

num = 0;

den = 1;

return;

}

while (tnum != 0)

{

if (tnum < tden)

{

temp = tnum;

tnum = tden;

tden = temp;

}

tnum = tnum - tden;

}

gcd = tden;

num = num / gcd;

den = den / gcd;

}

fraction fraction::operator + (fraction f)

{

int num1 = num\*f.den + den\*f.num;

int den1 = den\*f.den;

lowterms();

return fraction(num1,den1);

}

fraction fraction::operator - (fraction f)

{

int num1 = num\*f.den - den\*f.num;

int den1 = den\*f.den;

lowterms();

return fraction(num1,den1);

}

fraction fraction::operator \* (fraction f)

{

int num1 = num\*f.num;

int den1 = den\*f.den;

lowterms();

return fraction(num1,den1);

}

fraction fraction::operator / (fraction f)

{

int num1 = num\*f.den;

int den1 = den\*f.num;

lowterms();

return fraction(num1,den1);

}

int main()

{

fraction a,b,c,d,e,f;

char t;

cout<<"Enter the first fraction: "; a.get\_frac();

cout<<"Enter an operator(+, -, \*, /): "; cin>>t;

cout<<"Enter the second fraction: "; b.get\_frac();

switch(t)

{

case '+':

c=a+b;

cout<<"Addition: "; a.display(); cout<<" + "; b.display(); cout<<" = "; c.display();

cout<<endl;

break;

case '-':

d=a-b;

cout<<"Subtraction: "; a.display(); cout<<" - "; b.display(); cout<<" = "; d.display();

cout<<endl;

break;

case '\*':

e=a\*b;

cout<<"Multiplication: "; a.display(); cout<<" \* "; b.display(); cout<<" = "; e.display();

cout<<endl;

break;

case '/':

f=a/b;

cout<<"Division: "; a.display(); cout<<" / "; b.display(); cout<<" = "; f.display();

cout<<endl;

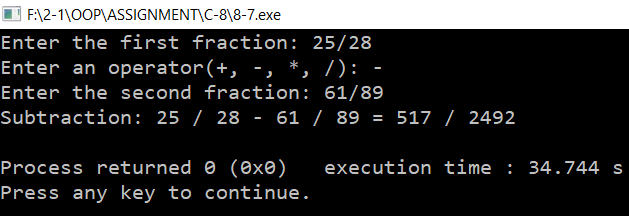
break;

}

return 0;

}

**Output:**

****

**Problem No.-09**

**Code:**

#include <iostream>

#include <process.h>

using namespace std;

const int LIMIT = 100;

class safearay

{

private:

int arr[LIMIT];

public:

int& operator [](int n)

{

if( n< 0 || n>=LIMIT )

{

cout << "\nIndex out of bounds";

exit(0);

}

return arr[n];

}

};

int main()

{

safearay sa1;

int x,y;

cout<<"Enter the array range: ";

cin>>x>>y;

for(int j=x; j<=y; j++)

sa1[j] = j\*10;

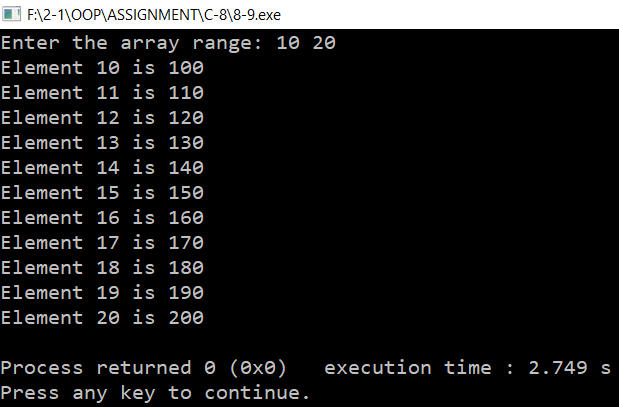
for(int j=x; j<=y; j++)

cout << "Element " << j << " is " <<sa1[j]<< endl;

return 0;

}

**Output:**

****

**Problem No.-11**

**Code:**

#include<iostream>

#include<iomanip>

using namespace std;

class Sterling

{

private:

long long int pounds;

int shilings,pence;

public:

Sterling():pounds(0),shilings(0),pence(0)

{ }

Sterling(double p)

{

pounds =static\_cast<int>(p) ;

double p1= p - pounds;

p1 = p1\*20.0;

shilings = static\_cast<int>(p1);

double s1 = p1-shilings;

s1=s1\*240.0 ;

pence = static\_cast<int>(s1);

}

Sterling(long long int pn, int sh,int pe):pounds(pn),shilings(sh),pence(pe)

{ }

void getSterling()

{

char ch;

cout<< "Enter an amount in pounds, shillings, and pence, format \x9c 9.19.11: ";

cin>>ch>>pounds>>ch>>shilings>>ch>>pence;

}

void putSterling()

{

cout<<'\x9c'<<pounds<<'.'<<shilings<<'.'<<pence<<endl;

}

Sterling operator + (Sterling);

Sterling operator - (Sterling);

Sterling operator \* (Sterling);

Sterling operator / (Sterling);

operator double();

};

Sterling Sterling::operator+(Sterling s2)

{

return Sterling(double(Sterling(pounds,shilings,pence))+double(s2));

}

Sterling Sterling::operator - (Sterling s2)

{

return Sterling(double(Sterling(pounds,shilings,pence))-double(s2));

}

Sterling Sterling::operator\*(Sterling s2)

{

return Sterling(double(Sterling(pounds,shilings,pence))\*double(s2));

}

Sterling Sterling::operator/(Sterling s2)

{

return Sterling(double(Sterling(pounds,shilings,pence))/double(s2));

}

Sterling::operator double()

{

double x= pounds + shilings/20 + pence/(12\*20);

return x;

}

int main()

{

Sterling s1, s2, sum, sub, mul,div;

Sterling n(4.4) ;

s1.getSterling();

s1.putSterling();

s2.getSterling();

s2.putSterling();

cout<<endl<<"Summation: ";

sum = s1 + s2;

sum.putSterling();

cout<<endl<<"Subtraction: ";

sub = s1 - s2;

sub.putSterling();

cout<<endl<<"Multiplication: ";

mul = s1 \* s2;

mul.putSterling();

mul = s1\*n;

mul.putSterling();

cout<<endl<<"Division: ";

div = s1 / s2 ;

div.putSterling();

div = s1/n;

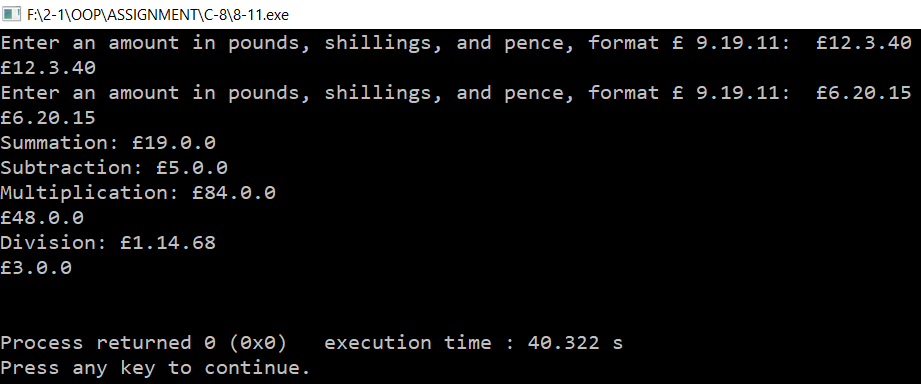
div.putSterling();

cout<<endl;

return 0;

}

**Output:**

****

**Chapter-09**

**Problem No.-01**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class publication

{

private:

string title;

float price;

public:

void getdata()

{

cout<<"Enter title: ";

cin>>title;

cout<<"Enter price: ";

cin>>price;

}

void putdata()

{

cout<<"Title: "<<title<<endl;

cout<<"Price: "<<price<<endl;

}

};

class book:public publication

{

private:

int count;

public:

void getdata()

{

publication::getdata();

cout<<"Enter number of pages: ";

cin>>count;

}

void putdata()

{

publication::putdata();

cout<<"Pages: "<<count<<endl;

}

};

class tape:public publication

{

private:

float time;

public:

void getdata()

{

publication::getdata();

cout<<"Enter playing time: ";

cin>>time;

}

void putdata()

{

publication::putdata();

cout<<"Time: "<<time;

}

};

int main()

{

book x;

tape y;

x.getdata();

y.getdata();

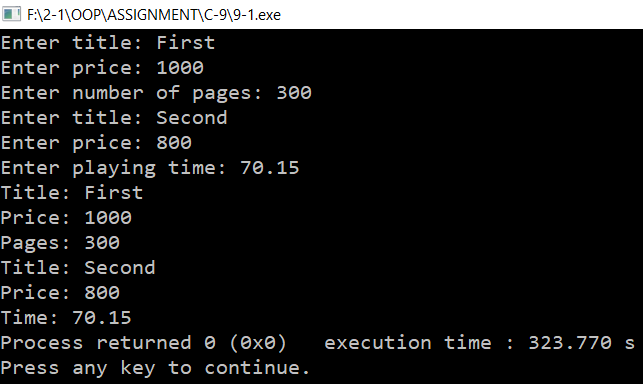
x.putdata();

y.putdata();

return 0;

}

**Output:**

****

**Problem No.-03**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class publication

{

private:

string title;

float price;

public:

void getdata()

{

cout<<"Enter title: "; cin>>title;

cout<<"Enter price: ";

cin>>price;

}

void putdata()

{

cout<<"Title: "<<title<<endl;

cout<<"Price: "<<price<<endl;

}

};

class sales

{

private:

float ara[5];

public:

void getdata()

{

cout<<"Enter sales: ";

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

{

cout<<i+1<<". ";

cin>>ara[i];

}

}

void putdata()

{

cout<<"Sales: ";

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

{

cout<<i+1<<". "<<ara[i]<<endl;

}

}

};

class book:public publication, public sales

{

private:

int count;

public:

void getdata()

{

publication::getdata();

cout<<"Enter number of pages: ";

cin>>count;

sales::getdata();

}

void putdata()

{

publication::putdata();

cout<<"Pages: "<<count<<endl;

sales::putdata();

}

};

class tape:public publication,public sales

{

private:

float time;

public:

void getdata()

{

publication::getdata();

cout<<"Enter playing time: "; cin>>time;

sales::getdata();

}

void putdata()

{

publication::putdata();

cout<<"Time: "<<time<<endl;

sales::putdata();

}

};

int main()

{

book x;

tape y;

x.getdata();

y.getdata();

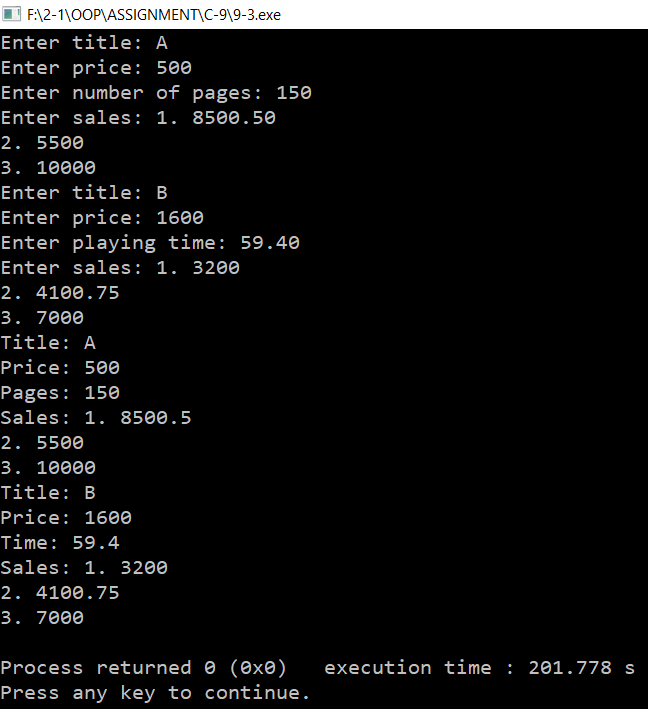
x.putdata();

y.putdata();

return 0;

}

**Output:**

****

**Problem No.-05**

**Code:**

#include <iostream>

using namespace std;

const int LEN = 80;

class employee

{

private:

char name[LEN];

unsigned long number;

public:

void getdata()

{

cout << "\n Enter last name: ";

cin >> name;

cout << " Enter number: ";

cin >> number;

}

void putdata() const

{

cout << "\n Name: "<< name;

cout << "\n Number: "<< number<<endl;

}

};

class employee2:public employee

{

private:

double compensation;

enum type{hourly,weekly,monthly};

type em;

public:

void getdata()

{

char ch;

employee::getdata();

cout<<"Enter compensation: ";

cin>>compensation;

cout<<"Enter payment type(h/w/m): ";

cin>>ch;

if(ch=='h' || ch=='H')

em=hourly;

else if(ch=='w' || ch=='W')

em=weekly;

else if(ch=='m' || ch=='M')

em=monthly;

}

void putdata() const

{

employee::putdata();

cout<<"Employee compensation: "<<compensation<<endl;

switch(em)

{

case hourly:

cout<<"Hourly"<<endl;

break;

case weekly:

cout<<"Weekly"<<endl;

break;

case monthly:

cout<<"Monthly"<<endl;

break;

}

}

};

class manager : public employee2

{

private:

char title[LEN];

double dues;

public:

void getdata()

{

employee2::getdata();

cout << " Enter title: ";

cin >> title;

cout << " Enter golf club dues: ";

cin >> dues;

}

void putdata() const

{

employee2::putdata();

cout << "\n Title: "<< title;

cout << "\n Golf club dues: "<< dues<<endl;

}

};

class scientist : public employee2

{

private:

int pubs;

public:

void getdata()

{

employee2::getdata();

cout << " Enter number of pubs: ";

cin >> pubs;

}

void putdata() const

{

employee2::putdata();

cout << "\n Number of publications: "<< pubs<<endl;

}

};

class laborer : public employee2

{ };

int main()

{

manager m1, m2;

scientist s1;

laborer l1;

cout << endl;

cout << "\nEnter data for manager 1";

m1.getdata();

cout << "\nEnter data for manager 2";

m2.getdata();

cout << "\nEnter data for scientist 1";

s1.getdata();

cout << "\nEnter data for laborer 1";

l1.getdata();

cout << "\nData on manager 1";

m1.putdata();

cout << "\nData on manager 2";

m2.putdata();

cout << "\nData on scientist 1";

s1.putdata();

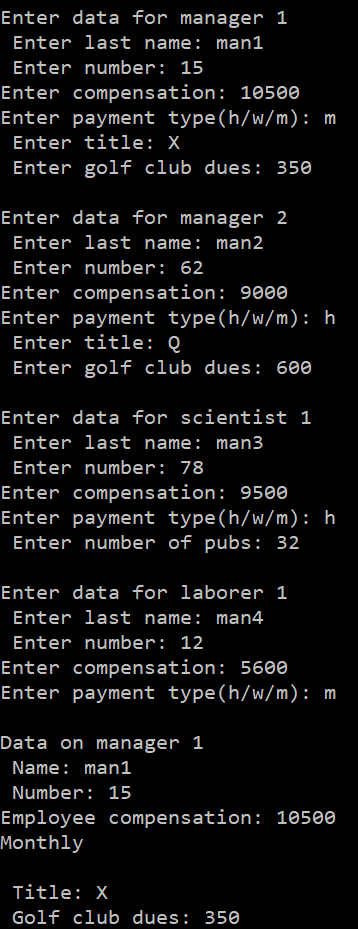
cout << "\nData on laborer 1";

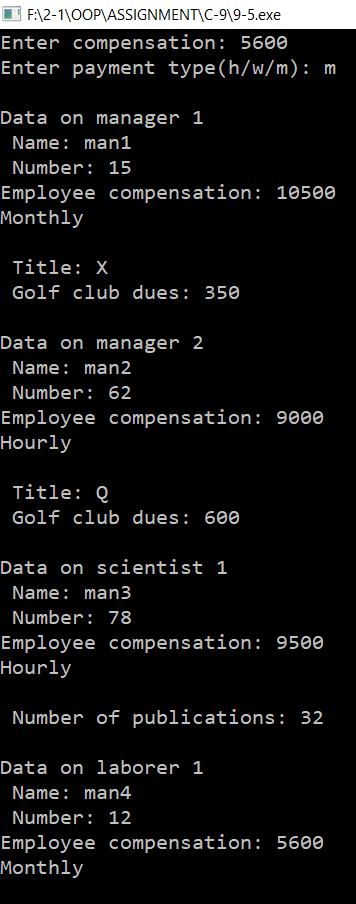
l1.putdata();

cout << endl;

return 0;

}

**Output:**

****

**Problem No.-07**

**Code:**

#include <iostream>

using namespace std;

class Counter

{

protected:

unsigned int count;

public:

Counter() : count()

{ }

Counter(int c) : count(c)

{ }

unsigned int get\_count() const

{

return count;

}

Counter operator ++ ()

{

return Counter(++count);

}

};

class CountDn : public Counter

{

public:

CountDn() : Counter()

{ }

CountDn(int c) : Counter(c)

{ }

CountDn operator -- ()

{

return CountDn(--count);

}

};

class CountUp:public CountDn

{

public:

CountUp():CountDn()

{ }

CountUp(int c):CountDn(c)

{ }

CountUp operator ++(int)

{

return CountUp(count++);

}

CountUp operator --(int)

{

return CountUp(count--);

}

};

int main()

{

CountDn c1;

CountUp c2(100);

cout << "\nc1=" << c1.get\_count();

cout << "\nc2=" << c2.get\_count();

++c1;

++c1;

c2++;

cout << "\nc1=" << c1.get\_count();

cout << "\nc2=" << c2.get\_count();

--c1;

c2--;

cout << "\nc1=" << c1.get\_count();

cout << "\nc2=" << c2.get\_count();

CountDn c3 = c2--;

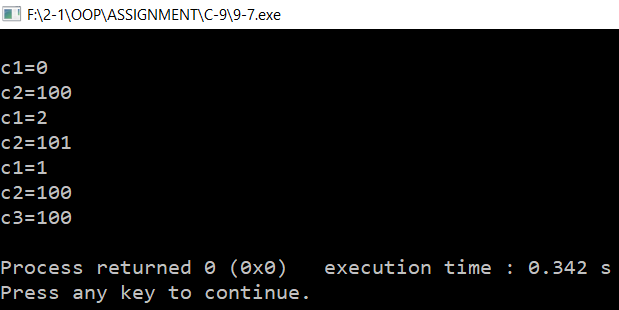
cout << "\nc3=" << c3.get\_count();

cout << endl;

return 0;

}

**Output:**

****

**Problem No.-09**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class publication

{

private:

string title;

float price;

public:

void getdata()

{

cout<<"Enter title: "; cin>>title;

cout<<"Enter price: "; cin>>price;

}

void putdata()

{

cout<<"Title: "<<title<<endl;

cout<<"Price: "<<price<<endl;

}

};

class date

{

private:

int month,day, year;

char ch;

public:

void get()

{

cout <<"Enter date in mm/dd/yy format: ";

cin >>month >>ch>>day >>ch>>year;

}

void display()

{

cout <<"Date: " <<month <<"/" <<day <<"/" <<year<<endl;

}

};

class publication2:public publication,public date

{

public:

void getin()

{

publication::getdata();

date::get();

}

void putin()

{

publication::putdata();

date::display();

}

};

class book:public publication2

{

private:

int count;

public:

void getdata()

{

publication2::getin();

cout<<"Enter number of pages: "; cin>>count;

}

void putdata()

{

publication2::putin();

cout<<"Pages: "<<count<<endl;

}

};

class tape:public publication2

{

private:

float time;

public:

void getdata()

{

publication2::getin();

cout<<"Enter playing time: "; cin>>time;

}

void putdata()

{

publication2::putin(); cout<<"Time: "<<time;

}

};

int main()

{

book x;

tape y;

x.getdata();

y.getdata();

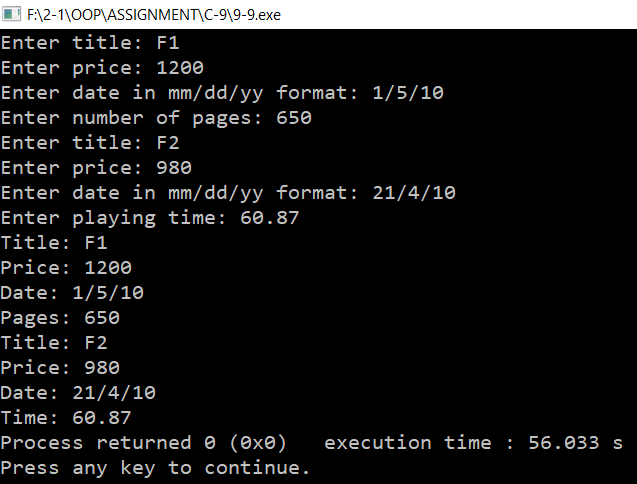
x.putdata();

y.putdata();

return 0;

}

**Output:**

****

**Problem No.-11**

**Code:**

#include <iostream>

using namespace std;

class Pair

{

protected:

enum { MAX = 50};

int st[MAX],top;

public:

Pair()

{

top = -1;

}

void push(int var)

{

st[++top] = var;

}

int pop(int v)

{

cout<<st[top--]<<")"<<" ";

}

};

class Stack2 : public Pair

{

public:

Stack2():Pair()

{ }

void push(int x, int y)

{

st[++top]=x; Pair::push(y);

}

int pop(int p, int q)

{

cout<<"("<<st[top--]<<","; Pair::pop(q);

}

};

int main()

{

Stack2 s1;

cout<<"Push: "<<endl;

s1.push(2,5); s1.push(7,1); s1.push(3,12);

cout<<"Popping: "<<endl;

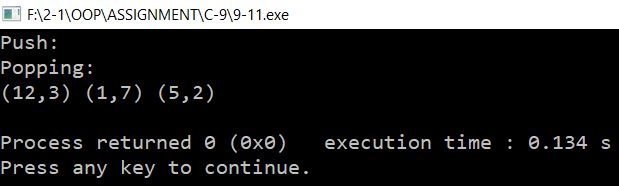
s1.pop(3,12); s1.pop(7,1); s1.pop(2,5);

cout << endl;

return 0;

}

**Output:**

****

**Chapter-10**

**Problem No.-01**

**Code:**

#include <iostream>

using namespace std;

int main()

{

float flarr[100];

char ch;

int num = 0;

do

{

cout << "Enter number: ";

cin >> \*(flarr+num++);

cout << " Enter another (y/n)? ";

cin >> ch;

}

while(ch != 'n');

float total = 0.0;

for(int k=0; k<num; k++)

total += \*(flarr+k);

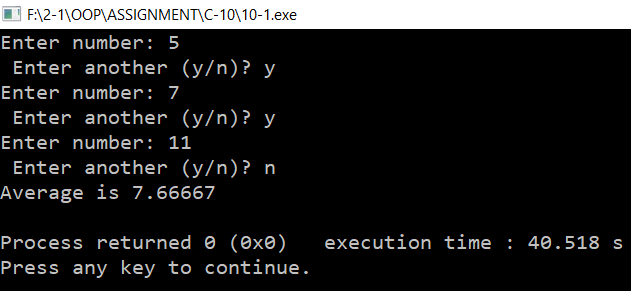
float average = total / num;

cout << "Average is "<< average << endl;

return 0;

}

**Output:**

****

**Problem No.-03**

**Code:**

#include <iostream>

#include <cstring>

using namespace std;

const int DAYS = 7;

int main()

{

void bsort(char\*\*, int);

char\* arrptrs[DAYS] = { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday","Friday", "Saturday"};

cout << "\nUnsorted: \n";

for(int j=0; j<DAYS; j++)

cout << \*(arrptrs+j) << endl;

bsort(arrptrs, DAYS); cout << "\nSorted: \n";

for(int j=0; j<DAYS; j++)

cout << \*(arrptrs+j) << endl;

return 0;

}

void bsort(char\*\* pp, int n)

{

void order(char\*\*, char\*\*);

int j, k;

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

for(k=j+1; k<n; k++)

order(pp+j, pp+k);

}

void order(char\*\* pp1, char\*\* pp2)

{

if( strcmp(\*pp1, \*pp2) > 0)

{

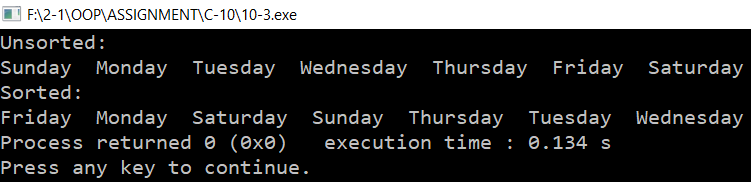
char\* tempptr = \*pp1; \*pp1 = \*pp2;

\*pp2 = tempptr;

}

}

**Output:**

****

**Problem No.-05**

**Code:**

#include <iostream>

#include <iomanip>

using namespace std;

void sum(int [], int [], int [], int);

int main()

{

int n= 20, ara1[n] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,18,20};

int ara2[n] = {11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30}, ara3[n] = {};

cout << setw(7) << "Array 1" << setw(3) << "+" << setw(10)<< "Array 2" << setw(3) << "=" << setw(10) << "Array 3" << endl;

sum(ara1,ara2,ara3,n);

}

void sum(int a[], int b[], int c[], int l)

{

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

{

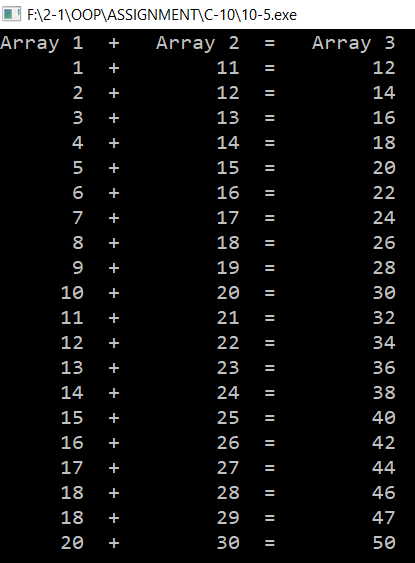
c[i]= b[i] + a[i];

cout << setw(7) << a[i] << setw(3) << "+" << setw(10)<< b[i] << setw(3) << "=" << setw(10) << c[i] << endl;

}

}

**Output:**

****

**Problem No.-07**

**Code:**

#include <iostream>

#include <string>

using namespace std;

class person

{

protected:

string name;

float salary;

public:

void setData()

{

cout << "Enter name: "; cin >> name;

cout<<"Enter salary: "; cin>>salary;

}

void printData()

{

cout << endl << name;

cout << endl << salary;

}

float getSalary()

{

return salary;

}

};

int main()

{

void salsort(person\*\*, int);

person\* persPtr[100];

person\* salptr[100];

int n = 0;

char choice;

do

{

salptr[n]=new person;

salptr[n]->setData();

n++;

cout << "Enter another (y/n)? ";

cin >> choice;

}

while( choice=='y' );

cout << "\nUnsorted list: ";

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

{

salptr[j]->printData();

}

salsort(salptr, n);

cout << "\nSorted list: ";

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

salptr[j]->printData();

cout << endl;

return 0;

}

void salsort(person\*\* pp, int n)

{

void order(person\*\*, person\*\*);

int j, k;

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

for(k=j+1; k<n; k++)

{

if((\*(pp+j))->getSalary() > (\*(pp+k))->getSalary() )

{

person\* tempptr=\*(pp+j);

\*(pp+j)=\*(pp+k);

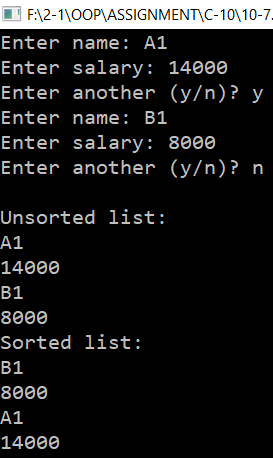
\*(pp+k)=tempptr;

}

}

}

**Output:**

****

**Problem No.-11**

**Code:**

#include<iostream>

#include<string>

#define MAX 100

using namespace std;

class Array

{

protected:

int a[10];

public:

void getData()

{

for(int i=0; i<10; i++)

cin>>a[i];

}

void ShowData()

{

for(int i=0; i<10; i++)

cout<<a[i]<<" ";

}

int& operator[](int n)

{

if(n>MAX)

cout<<"Index Out Of Boundary!"<<endl;

return a[n];

}

};

int main()

{

Array\* ara[MAX];

char ch; int n=0;

do

{

ara[n]=new Array;

cout<<"Enter data: "; ara[n]->getData();

n++;

cout<<endl<<"Continue?(y/n) "; cin>>ch;

}

while(ch!='n');

cout<<"Entered array: ";

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

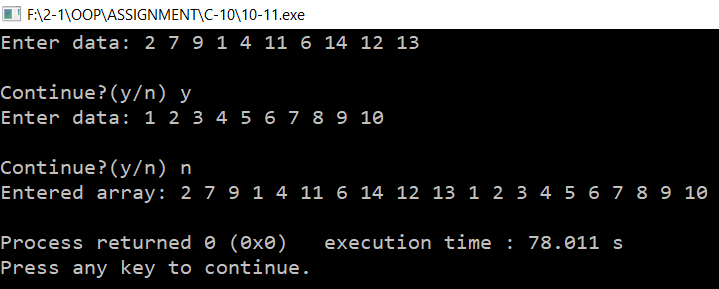
ara[i]->ShowData();

cout<<endl;

return 0;

}

**Output:**

****