LAB_PROGRAM_7

Write a C++ program to overload unary operator ++ and – to work with counter class object. The counter class should have one data member as count.

// Sample Input

11

//Sample Output

12

13

-13

```
class counter
{
    int count;
    public:
    counter();
    counter(int);
    void operator ++();
    void operator ++(int);
    void operator -();
    void show();
};
counter::counter()
{
    count=0;
}

counter::counter(int a)
{
    count=a;
}
void counter::operator ++()
{
    count++;
   // cout<<count<<endl;
}
void counter::operator ++(int a)
{
    count++;
   // cout<<count<<endl;
}
void counter::operator -()
{
    count=-count;
}
void counter::show()
{
    cout<<count<<endl;
}

int main() {↔}
```

Sum of Distance

You are given a task to perform addition of two distance which are given in kilometer(km) and meter(m). Also check if sum of meters is more than 1000, then you need to increment kilometers.

Sample Input
2 //kilometer of 1st distance
100 //meter of 1st distance
3 //kilometer of 2nd distance
500 //meter of 2nd distance

Sample Output
5 //sum of kilometer(2 and 3)
600 //sum of meter(100 and 500)

```
#include ↔

using namespace std;
class sum
{
    int km,m;
    public:
    sum();
    void getdata();
    void showdata();
    sum operator +(sum &);
};
sum::sum()
{
   km=0;m=0;
}
void sum:: getdata()
{
    cin>>km>>m;
}
void sum::showdata()
{
    cout<<km<<endl<<m;
}
sum sum::operator +(sum &r)
{
    r.km=km+r.km;
    r.m=m+r.m;
    if(r.m>=1000)
    {
        r.m=r.m-1000;
        r.km++;
    }
    return r;
}

int main()
{
    sum d1,d2,d3;
    d1.getdata();
    d2.getdata();
    d3=d1+d2;
    d3.showdata();
    return 0;
}
```

LAB_PROGRAM_8

Write a C++ program to add, subtract and multiply two matrices using operator overloading.

Sample Input :
2 3 4 // First Matrix
4 5 6
1 4 2

4 8 8 // Second matrix
1 1 1
1 1 1
Sample Output :
6 11 12 //Addition of Matrices
5 6 7
2 5 3

-2 -5 -4 //Subtraction of Matrices
3 4 5
0 3 1

15 23 23 //Product of Matrices
27 43 43
10 14 14

```
mat::mat()
{
    a[3][3]=0;
}
void mat::getdata()
{
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
            cin>>a[i][j];
    }
}
mat mat::operator +(mat r)
{mat w;
       for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
         w.a[i][j]=a[i][j]+r.a[i][j];
    }
    return w;
}
mat mat::operator -(mat r)
{mat w;
       for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
          w.a[i][j]=a[i][j]-r.a[i][j];
    }
    return w;
}
mat mat::operator *(mat r)
{
   mat q;
       for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            q.a[i][j]=0;
            for(int k=0;k<3;k++)
                q.a[i][j]=q.a[i][j]+a[i][k]*r.a[k][j];

        }

    }
    return q;
}
void mat::disp()
{
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
            cout<<a[i][j]<<" ";
        cout<<endl;
    }
}
```

LAB_PROGRAM_9

Alisha always gets confused with the operators used with strings. When she has to concatenate two strings suppose string1 and string2, she used to write string1+string2. And when she has to compare whether the two strings are equal or first string is greater or smaller than string 2 again she writes like string1==string2,string1>strin2 and string1<string2. Which gives the error in program as these operators does not work with strings directly. So, help out her to work these operators with strings directly.

Sample Input 1 :

C++ // string 1
Programming // string 2

Sample Output 1 :

Concatenated strings are : C++Programming
Are the strings equal : Both are unequal
Is string1 smaller than string2 : Yes
Is string1 larger than string2 : No

Sample Input 2 :
Program
Program

Sample Output 2 :

Concatenated strings are : C++Program Learning
Are the strings equal : Both are equal
Is string1 smaller than string2 : No
Is string1 larger than string2 : No

```
using namespace std;
class strings
{
    char str[100];
    public:
    strings(){}
    void read();
    void operator +(strings);
    void operator ==(strings);
    void operator >(strings);
    void operator <(strings);
};

/* Enter your code here. Read input from STDIN. Print output to STDOUT */
void strings::read()
{
    cin.getline(str,100);
}
void strings::operator +(strings r)
{
   // cout<<"Concatenated strings are:";
    cout<<str<<r.str<<endl;
}
void strings::operator <(strings r)
{

if(strcmp(str,r.str)<0)
    cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

}
void strings::operator ==(strings r)
{

if(strcmp(str,r.str)==0)
    cout<<"Both are equal"<<endl;
    else
        cout<<"Both are unequal"<<endl;

}
void strings::operator >(strings r)
{

if(strcmp(str,r.str)>0)
    cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

}

int main()
{
    strings string1,string2;
    string1.read();
```

Type_Conversion

Write a program to demonstrate the use of type conversions with Distance class.

Sample Input :
1234 //distance in metres

Sample Output :
1.234 //distance in kilometres

Original code | C++

```
#include ⇔
using namespace std;
class Distance
{
    int km,m;
public:
    Distance()
    {
        km=0;
        m=0;
    }
    Distance(int);
    operator float();
};

 Distance::Distance(int r)
 {
     m=r;
 }
Distance::operator float()
{
      float k=m*0.001;
 //   cout<<k;
    return k;
}

int main()
{
    int dist;
    cin>>dist;
    Distance d1;
    d1=dist;
    float k=d1;
    cout<<k;
    return 0;
}
```

Line: 37 Col: 1

Test against custom input | Run Code | Submit code & Continue

(You can submit any number of times)

☆ OVERLOADING +,- AND * OPERATOR

Write a program to overload +, - and * for matrix addition,subtraction and multiplication.

Sample Input

```
2 2 2
3 3 3
1 1 1
1 1 1
1 1 1
1 1 1
```

Sample Output

```
Addition of two matrices
3 3 3
4 4 4
2 2 2
Subtraction of two matrices
1 1 1
2 2 2
0 0 0
Multiplication of two matrices
6 6 6
9 9 9
3 3 3
```

Explanation

Sample Input

First three lines are the input for first matrix
next three lines are the input for second matrix

Sample Output

The result of addition, subtraction and multiplication of two matrices given in sample input
each element is separated by space in each row

```
Matrix()
{
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
            cin>>mat[i][j];
    }
}
Matrix operator +(Matrix r)
{
  for(int i=0;i<3;i++)
  {
      for(int j=0;j<3;j++)
          r.mat[i][j]=mat[i][j]+r.mat[i][j];
  }
   return r;
}
void Show()
{
   cout<<endl;
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            cout<<mat[i][j]<<" ";
        }
        if(i!=2)
        cout<<endl;
    }
}
Matrix operator -(Matrix r)
{
     for(int i=0;i<3;i++)
  {
      for(int j=0;j<3;j++)
          r.mat[i][j]=mat[i][j]-r.mat[i][j];
  }
   return r;
}
Matrix operator *(Matrix r)
{
   Matrix q;
       for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            q.mat[i][j]=0;
            for(int k=0;k<3;k++)
                q.mat[i][j]=q.mat[i][j]+mat[i][k]*r.mat[k][j];

        }

    }
    return q;
}
};
```

LINK-2C++2018_Operator Ove | LINK-2C++2018_Operator Ove

https://www.hackerrank.com/tests/86ef5aejh4t/questions/6ook29imp99 60%

You must log in to this network before you can access the Internet. Open Network Login Page

LINK-2C++2018_Operator Overloading and Type Conversion | 28d 19h to test end | 28/37 Attempted | Kashish Chaudhary

☆ COMPLEX NUMBERS

Write a program to overload following operators +, -, and * to perform addition, subtraction,and multiplication on given complex numbers.

Sample Input

4 6
2 1
1 2
3 2

Sample Output

7-1i

Explanation

Sample Input : Four complex numbers are given
In first line, first number is the real part and 2nd number is the imaginary part of first complex number.
In second line, first number is the real part and 2nd number is the imaginary part of second complex number.
In third line, first number is the real part and 2nd number is the imaginary part of third complex number.
In forth line, first number is the real part and 2nd number is the imaginary part of forth complex number.

Sample Output

Result of C1+C2-C3*C4 means (4+6i)+(2+1i)-(1+2i)*(3+2i)

Original code | C++

```
#include ↔

using namespace std;
class complex
    {
   int r,i;
    public:

complex()
{
    cin>>r>>i;
}
complex operator +(complex w)
{
    complex q;
    q.r=r+w.r;
    q.i=i+w.i;
    return q;

}
complex operator -(complex w)
{
    complex q;
    q.r=r-w.r;
    q.i=i-w.i;
    return q;

}
complex operator *(complex w)
{
    complex q;
    q.r=(r*w.r)+((-1)*(i*w.i));
    q.i=(r*w.i)+(w.r*i);
    return q;

}
void show()
{
    if(i<0)
        cout<<r<<i<<"i";
    else
        cout<<r<<"+"<<i<<"i";
}
};

int main() {↔}

```

Line: 30 Col: 1

ENG INTL 5:09 PM 29-Aug-18

Addition Through Friend

Write a program to overload +,-, * and / to add, subtract, multiply and divide two complex numbers

Sample Input

4 6
2 3
2
3

Sample Output

6+9i
6+8i
7+9i

Sample Input

2 1
3 4
5
1

Sample Output

5+5i
7+6i
3+2i

Explanation

Sample Input

First two lines are the two complex numbers
next two lines are the two integer numbers

Sample Output

First line: Addition of two complex numbers
Second line: First complex number + first integer
Third line: First complex number + second integer

```
23
24 using namespace std;
25 class complex
26     {
27     int real,imag;
28     public:
29     friend complex operator+(int,complex);
30     friend complex operator+(complex,int);
31     friend complex operator +(complex,complex);
32     void show();
33     void set();
34
35 };
36
37 void complex::set()
38 {
39     cin>>real>>imag;
40 }
41 complex operator +(complex r,complex w)
42 {
43     complex q;
44     q.real=w.real+r.real;
45     q.imag=w.imag+r.imag;
46     return q;
47 }
48 complex operator +(complex w,int r)
49 {
50     complex q;
51     q.real=w.real+r;
52     q.imag=w.imag+r;
53     return q;
54 }
55 complex operator +(int r,complex w)
56 {
57     complex q;
58     q.real=w.real+r;
59     q.imag=w.imag+r;
60     return q;
61 }
62 void complex::show()
63 {
64
65     if(imag>=0)
66         cout<<real<<"+"<<imag<<"i";
67     else
68        cout<<real<<imag<<"i";
69     cout<<endl;
70 }
71
72 int main() {↔}
87
88
89
```

LINK-2C++2018_Operator Over

LINK-2C++2018_Operator Over

https://www.hackerrank.com/tests/86ef5aejh4t/questions/6h61b06amih 60%

You must log in to this network before you can access the Internet. Open Network Login Page

LINK-2C++2018_Operator Overloading and Type Conversion | 28d 19h to test end | 28/37 Attempted | Kashish Chaudhary

FRIEND FUNCTION

For class **example**, declare display function outside the class and it should display the private data members of the example class.

SAMPLE INPUT:

```
13
```

SAMPLE OUTPUT:

```
13
```

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour

Original code | C++

```
#include ⇔

using namespace std;
class example
{
    int a;
public:
    void get();
     friend void display(example);
};


void display(example r)
{
    cout<<r.a;
}
void example::get()
{
    cin>>a;
}
int main()
{
    example x;
    x.get();
    display(x);
}

```

Line: 34 Col: 1

Test against custom input | Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

ENG INTL 5:10 PM 29-Aug-18

FUNCTION OUTSIDE THE CLASS

A function which is not a member function of any of the two classes wants to access the private data members of two classes with the help of one friend function and perform subtraction of data members of both the classes.

SAMPLE INPUT 1:

10 // member of first class

5 //member of second class

SAMPLE OUTPUT 1:

5

SAMPLE INPUT 2:

10 // member of first class

20 //member of second class

SAMPLE OUTPUT 2:

-10

Original code | C++

```
#include ↔

using namespace std;
class abc;
class ABC
{
    int a;

public:
void get()
{
    cin>>a;
}
friend void diff(ABC x,abc y);
};
class abc{
    int b;
    public:
    void get()
    {
        cin>>b;
    }
friend void diff(ABC x,abc y);
};
void diff(ABC x,abc y)
{
    int z;
    z=x.a-y.b;
    cout<<z;
}

int main()
{
    ABC obj1;
    abc obj2;
    obj1.get();
    obj2.get();
    diff(obj1,obj2);
    return 0;
}
```

Line: 30 Col: 1

Test against custom input | Run Code | Submit code & Continue

☆ FRIEND FUNCTION OF ONE CLASS AS A MEMBER FUNCTION OF ANOTHER CLASS

Write a program to declare the member function in one class and make it the friend of another class and perform subtraction between member function of both the classes.

SAMPLE INPUT 1:

10 // member of first class

5 //member of second class

SAMPLE OUTPUT 1:

5

SAMPLE INPUT 2:

10 // member of first class

20 //member of second class

SAMPLE OUTPUT 2:

-10

```
class abc;
class ABC
{
    int a;
public :
    void get();
    void display();
    void diff(abc);
};
class abc
{
    int a;
public :
    void get();
    void display();



friend void ABC::diff(abc);
};
void abc::get()
{
    cin>>a;
}
void ABC::get()
{
    cin>>a;
}
void ABC::diff(abc x)
{
    int w;
    w=a-x.a;
    cout<<w;
}
int main()
{
    ABC obj1;
    obj1.get();
    abc obj2;
    obj2.get();

obj1.diff(obj2);
    return 0;
}
```

Line: 43 Col: 1

Test against custom input

Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

LINK-2C++2018_Operator Over | LINK-2C++2018_Operator Over

https://www.hackerrank.com/tests/86ef5aejh4t/questions/c8rktcfjtci 60%

You must log in to this network before you can access the Internet. Open Network Login Page

LINK-2C++2018_Operator Overloading and Type Conversion | 28d 19h to test end | 28/37 Attempted | Kashish Chaudhary

String Overloading

Alisha always gets confused with the operators used with strings. When she has to compare whether the two strings are equal or first string is greater or smaller than string 2 , she writes like string1==string2 ,string1>string2 and string1<string2. Which gives the error in program as these operators does not work with strings directly. So, help out her to work these operators with strings directly.

Sample Input1
string1 = hello
string 2 = hello

Sample Output1
Both strings are equal

Sample Input2
string1 = hello
string2 = Hello

Sample Output2
String 1 is greater than string 2

Sample Input3
string1 = Hello
string2 = hello

Sample Output3
String 1 is smaller than string 2

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour

Original code | C++

```
#include ↔
using namespace std;
class str
    {
    char s[50];
    public:

void set()
{
    cin.getline(s,50);
}
int operator ==(str q)
{
    if(strcmp(s,q.s)==0)
        return 1;
    else
        return 0;
}
int operator <(str q)
{
    if(strcmp(s,q.s)<0)
        return 1;
    else
        return 0;
}
int operator >(str q)
{
    if(strcmp(s,q.s)>0)
        return 1;
    else
        return 0;
}
};

int main() {↔}
```

Line: 31 Col: 1

Test against custom input | Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

ENG INTL 5:11 PM 29-Aug-18

LINK-2C++2018_Operator Ove | LINK-2C++2018_Operator Ove

https://www.hackerrank.com/tests/86ef5aejh4t/questions/6h61b29m114 60%

You must log in to this network before you can access the Internet. Open Network Login Page

LINK-2C++2018_Operator Overloading and Type Conversion | 28d 19h to test end | 28/37 Attempted | Kashish Chaudhary

Program to implement dynamic constructors

Using dynamic constructors, initialize the elements of a 1D aray and then display the array.

Sample Input:
2 (Size of the array)
2 (Element1 of the array)
3 (Element2 of the array)

Sample Output:
23 (The array elements that were entered)

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour

Original code | C++

```
#include ↔

using namespace std;
class test
{
  int *arr;
  int n;

public:
test(int s)
{
    n=s;
    arr=new int[n];
    for(int i=0;i<n;i++)
        cin>>arr[i];
}
void show()
{
    for(int i=0;i<n;i++)
        cout<<arr[i];
}
};

int main()
{
  int size;
  //cout<<"Enter the size of the array :";
  cin>>size;
  //cout<<"Enter the array elements:";
  test t1(size);
  //cout<<"The array elements entered are :";
  t1.show();
  cout<<endl;
return 0;

}
```

Line: 30 Col: 1

Test against custom input | Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

ENG INTL 5:11 PM 29-Aug-18

LINK-2C++2018_Operator Over | LINK-2C++2018_Operator Over

https://www.hackerrank.com/tests/86ef5aejh4t/questions/5a3c9a20e4n 50%

You must log in to this network before you can access the Internet. Open Network Login Page

LINK-2C++2018_Operator Overloading and Type Conversion | 28d 19h to test end | 28/37 Attempted | Kashish Chaudhary

SORTING ALL DETAILS

An Accountant have list of N customer stored as private data that include account id and amount. He wants an application to sort the record according to any factor that may be account number or amount (i or a respectively) .Write a c++ program to help him.

Sample Input:

i 4 //i is a choice to sort data on the bases of account id and N=4 is total number of records
4 900 //read the account id and amount, N=4 times
3 800
2 1000
1 900

Sample Output:

1 900
2 1000
3 800
4 900

Note: If number of records are out of range then output should be "RECORD OUT OF RANGE" and in case of wrong choice output should be "WRONG CHOICE".

Constraint:

Two Choices i and a
0<n<=30

```
        }
};

void sort1::show()
{
    for(int i=0;i<n;i++)
        cout<<data[i][0]<<" "<<data[i][1]<<endl;
}
void ssort(sort1 & sn)
{int t,q;
 if(sn.n>0 && sn.n<=30)
 {
 if(sn.choice=='i')
 {
    for(int i=0;i<sn.n;i++)
    {
        for(int j=i+1;j<sn.n;j++)
        {
            if(sn.data[i][0]>sn.data[j][0])
            {
                t=sn.data[i][0];
                sn.data[i][0]=sn.data[j][0];
                sn.data[j][0]=t;


                  q=sn.data[i][1];
                sn.data[i][1]=sn.data[j][1];
                sn.data[j][1]=q;

            }
        }
    }

 }
 else if(sn.choice=='a')
 {
     for(int i=0;i<sn.n;i++)
    {
        for(int j=i+1;j<sn.n;j++)
        {
            if(sn.data[i][1]>sn.data[j][1])
            {
                t=sn.data[i][1];
                sn.data[i][1]=sn.data[j][1];
                sn.data[j][1]=t;



                q=sn.data[i][0];
                sn.data[i][0]=sn.data[j][0];
                sn.data[j][0]=q;

            }
        }
    }
 }
  else
      cout<<"WRONG CHOICE";
 }
 else
     cout<<"RECORD OUT OF RANGE";
}


int main()
{
```

ENG INTL 5:13 PM 29-Aug-18

ADD TWO POINTS

Overload the binary operator + to add two Point Type Objects.
The Point class has two members of integer datatype,
representing the coordinates of a point.

Sample Input:
1 2 //Coordinates of Point P1
1 2 //Coordinates of Point P2
Sample Output:
2 4 //Sum of the coordinates of the points P1 and P2

Original code | C++

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
class Point
{
    int x,y;
    public:
    Point()
    {
        x=0;
        y=0;
    }
    Point(int a,int b)
    {
        x=a;
        y=b;
    }
    Point operator +(Point a )
    {
        Point q;
        q.x=x+a.x;
        q.y=y+a.y;
        return q;
    }
    void putdata()
    {
        cout<<x<<" "<<y;
    }
};


int main()
{
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    Point p1(a,b),p2(c,d),p3;
```

ADD THE NUMBERS

Write a code to find out sum of two numbers x and y; where x and y are the private data members of class one and class two respectively.

Sample input

4 6

Sample output

10

Original code | C++

```
#include ⇔

using namespace std;
class two;
class one
{

float a;
public:
void setdata(int x)
{
    a=x;
}
friend void add(one q,two w);
};
class two
{
    float b;
    public:
    void setdata(int x)
{
    b=x;
}
    friend void add(one q,two w);

};
 void add(one q,two w)
 {
     float r;
     r=q.a+w.b;
     cout<<r;
 }

int main()
{
one a;
two b;
float i,j;
cin>>i>>j;

a.setdata(i);
b.setdata(j);
add(a,b);
return 0;
```

ADMISSION FEES

Three schools in chandigarh region have started their admission process.The main school out of three will decide the admission fee for particular class and rest two schools have to follow same fee structure for their schools too.
fee must be equal to or between 2000 -4000 otherwise print**"The school fee not affordable"**
The input will be name of class and fee for that class

Sample Input

nursery
2500

Sample output

For admission in smartwonder
The fee is Rs.2500for classnursery
For admission in kindergarten
The fee is Rs.2500for classnursery
For admission in kidscamp
The fee is Rs.2500for classnursery

Draft saved 05:14 pm

Original code | C++

```
#include ⇔

using namespace std;

class school
{
    char str[50];
    float w;
    static int i;
    public:
    school(char a[50],float b)
    {
        strcpy(str,a);
        w=b;
    }
    school(school &c)
    {
        strcpy(str,c.str);
        w=c.w;
    }
    void display()
    {
        if(w>=2000 && w<=4000)
        cout<<"The fee is Rs."<<w<<"for class"<<str<<endl;
        else if(i<1)
        {
            cout<<"The school fee not affordable";
            i++;
        }
    }
};
int school::i=0;


int main() {⇔}
```

Line: 26 Col: 1

LINK-2C++2018_Operator Over

LINK-2C++2018_Operator Over

https://www.hackerrank.com/tests/86ef5aejh4t/questions/ms5fj024at 70%

You must log in to this network before you can access the Internet. Open Network Login Page

LINK-2C++2018_Operator Overloading and Type Conversion | 28d 19h to test end | 28/37 Attempted | Kashish Chaudhary

STRING DECRYPT

A class **string1** allocates appropriate memory (using new operator) to a string **passed from the main function**, and then decrypt the string, by subtracting 3 to the ASCII value of each alphabet of the string.

Sample Input:

Deklodvkd

Sample Output:

Abhilasha

```
    private:
        int len;
        char *str;
    public:
        string1();
        string1(char *s);
        ~string1();
        void show_string1();
        void decrypt( );
};

string1::string1()
{
    str=0;
    len=0;
}
string1::string1(char a[100])
{
    len=strlen(a);
    str=new char[len+1];
    strcpy(str,a);
}
void string1::decrypt()
{
    for(int i=0;i<len;i++)
    {
        str[i]=str[i]-3;
    }

}
void string1::show_string1()
{
    for(int i=0;i<len;i++)
        cout<<str[i];
}
string1::~string1()
{

}

int main()
{
```

ENG INTL 5:15 PM 29-Aug-18

OPERATOR OVERLOADING

Create a class arradd which has an integer array as a data member.
Overload the + operator to **add two arradd type objects** and the result of addition should be stored in the third object.
Note: size of both the arrays is same and determined at run-time.

Sample Input:

```
5               //size of the arrays
1 2 3 4 5   //elements of first array
3 3 3 3 3   //elements of second array
```

Sample Output:

```
4   //result of addition
5   //of first and
6   //second array
7
8
```

Original code | C++

```
#include ↔

using namespace std;

int size;

class arradd
{
    int *arr;
    public:
    arradd()
    {
        arr=new int[size];
    }
    void getdata()
    {
        for(int i=0;i<size;i++)
            cin>>arr[i];
    }
    arradd operator +(arradd b)
    {
        arradd q;
        for(int i=0;i<size;i++)
        {
            q.arr[i]=arr[i]+b.arr[i];
        }
        return q;

    }
    void putdata()
    {
        for(int i=0;i<size;i++)
            cout<<arr[i]<<" ";
    }
};

int main()
{
cin>>size;

arradd A1,A2,A3;
```