KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

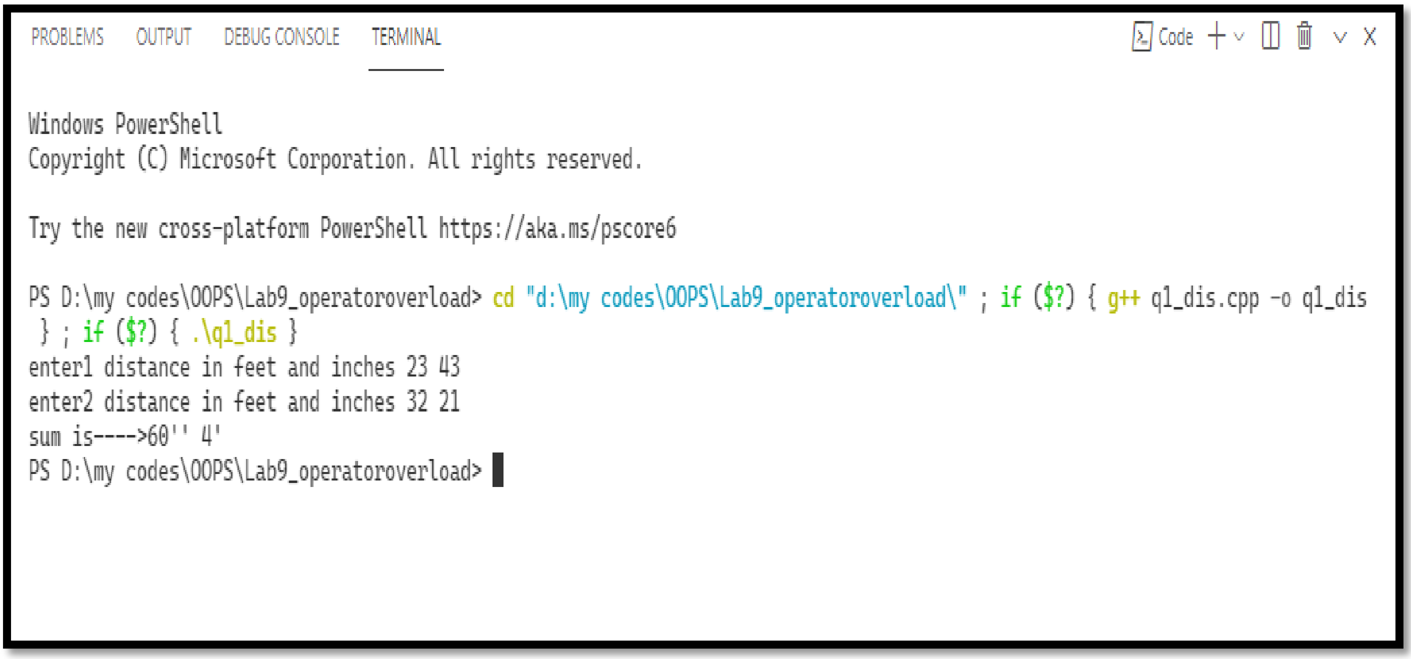
Deemed to be University U/S 3 of UGC Act, 1956

LAB 8-OPERATOR OVERLOADING

- Name :HITU RAJ
- Roll no. :2005025
- Branch :CSE

```
/*Q1A>.WAP to overload following operators for class distance, which 
stores the
distance in feet and inches.
Binary + to

-add two objects (D3=D1+D2)
-

Unary -*/

#include <bits/stdc++.h>
using namespace std;

class distance_025
{

    int feet_025;
    int inches_025;

public:
    void input()
    {

        cin >> feet_025 >> inches_025;
    }
    void display()
    {
        cout << "sum is---->" << feet_025 << "'' " << inches_025 <<
"'\n";
    }
    distance_025 operator+(distance_025);
};

distance_025 distance_025 ::operator+(distance_025 b)
{
    distance_025 t;

    t.feet_025 = feet_025 + b.feet_025;
    t.inches_025 = inches_025 + b.inches_025;
    if (t.inches_025 > 11)
    {
        t.feet_025 = t.feet_025 + t.inches_025 / 12;
        t.inches_025 = t.inches_025 % 12;
    }
    return t;
}
int main()
{
    distance_025 a[2], sum;

    for (int i = 0; i < 2; i++)
    {
        cout << "enter" << i + 1 << " distance in feet and inches ";

        a[i].input();
    }
    sum = a[0] + a[1];
    sum.display();

    return 0;
}
```

OUTPUT -1

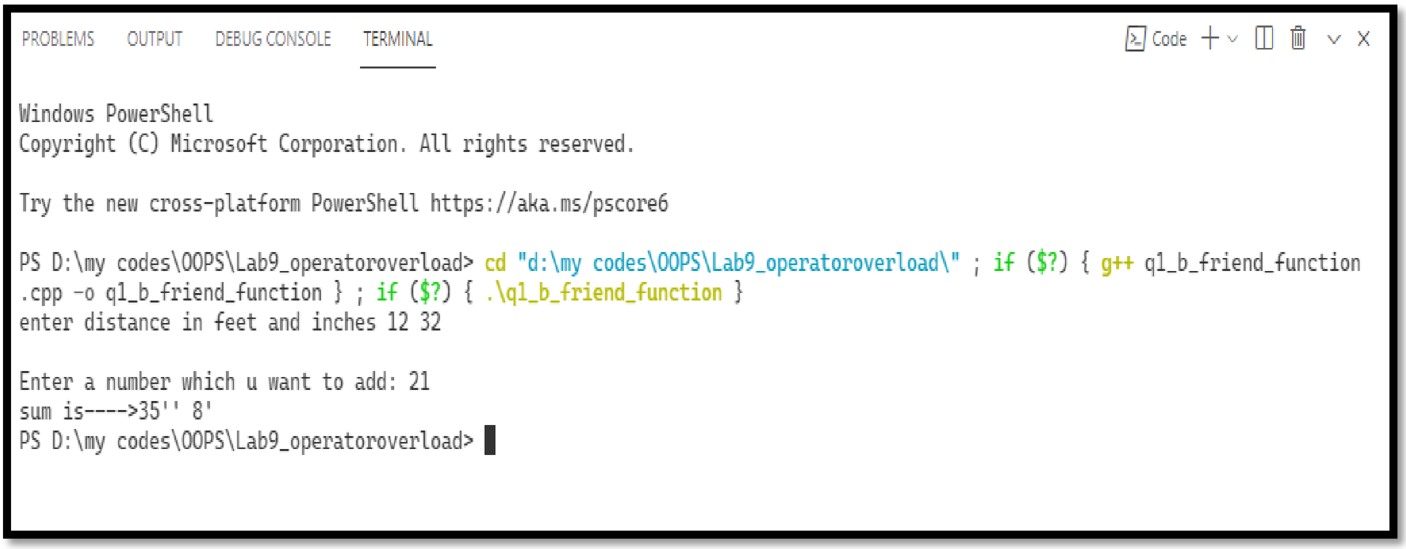


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q1_dis.cpp -o q1_dis
 } ; if ($?) { .\q1_dis }
enter1 distance in feet and inches 23 43
enter2 distance in feet and inches 32 21
sum is---->60'' 4'
PS D:\my codes\OOPS\Lab9_operatoroverload>
```

```
/*Q1B>Add an object to an integer, where the integer should be added
to the inches value ( D2=4+D1)*/


#include <bits/stdc++.h>
using namespace std;

class distance_025
{

    int feet_025;
    int inches_025;

public:
    void input()
    {

        cin >> feet_025 >> inches_025;
    }
    void display()
    {
        cout << "sum is---->" << feet_025 << "'' " << inches_025 <<
"'\n";
    }
 friend distance_025 operator +(int,distance_025);
};
```

```cpp
distance_025 operator +(int a, distance_025 b)
{
    distance_025 t;

    t.feet_025 = a + b.feet_025;
    t.inches_025 =  b.inches_025;
    if (t.inches_025 > 11)
    {
        t.feet_025 = t.feet_025 + t.inches_025 / 12;
        t.inches_025 = t.inches_025 % 12;
    }
    return t;
}
int main()
{
    distance_025 a, sum;
    int x;



        cout << "enter distance in feet and inches ";

        a.input();
        cout<<"\nEnter a number which u want to add: ";
        cin>>x;

    sum = x + a;
    sum.display();

    return 0;
}
```

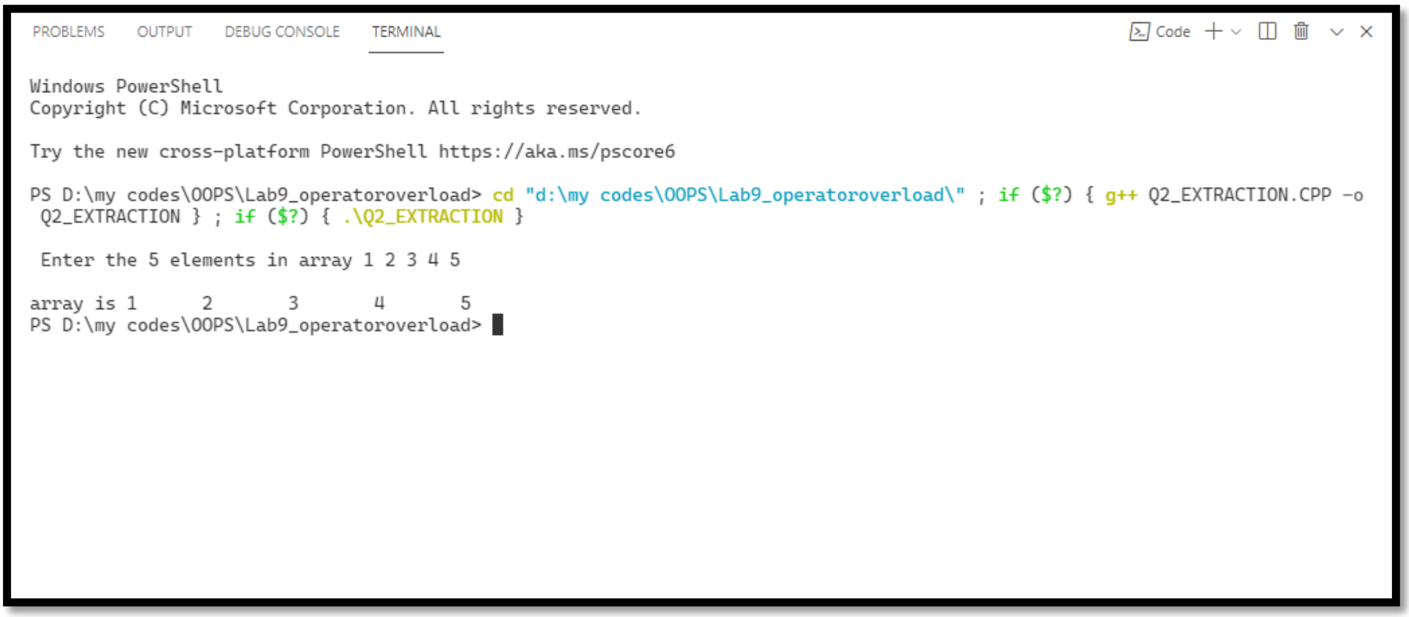
OUTPUT -1-b



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q1_b_friend_function
.cpp -o q1_b_friend_function } ; if ($?) { .\q1_b_friend_function }
enter distance in feet and inches 12 32

Enter a number which u want to add: 21
sum is---->35'' 8'
PS D:\my codes\OOPS\Lab9_operatoroverload>
```

```
/*2Create a class to store an integer array. Overload insertion and
extraction
operator to input and display the array elements.n*/
#include <iostream>
using namespace std;
class array
{

    int arr[5];

    friend void operator>>(istream &in, array &c)
    {
        cout << "\n Enter the 5 elements in array ";
        for (int i = 0; i < 5; i++)
        {
            in >> c.arr[i];
        }
    }
    friend void operator<<(ostream &out, const array &c)
    {
        cout << "\narray is ";
        for (int i = 0; i < 5; i++)
        {
            out << c.arr[i] << "\t";
        }
    }
};

int main()
{

    array a;
    cin >> a;
    cout << a;

    return 0;
}
```

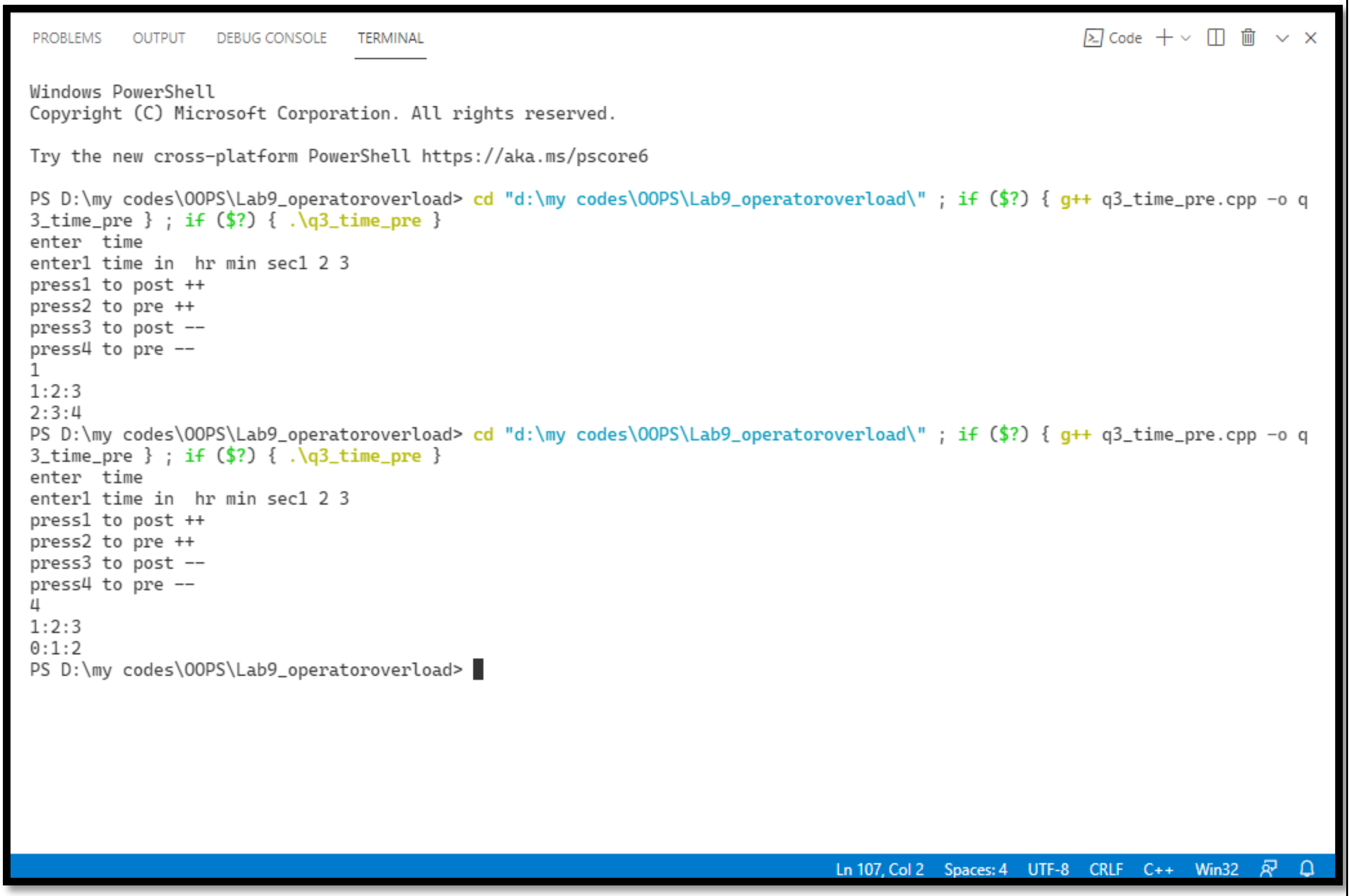
OUTPUT -2



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ Q2_EXTRACTION.CPP -o
 Q2_EXTRACTION } ; if ($?) { .\Q2_EXTRACTION }

 Enter the 5 elements in array 1 2 3 4 5

array is 1       2       3       4       5
PS D:\my codes\OOPS\Lab9_operatoroverload> 
```

```
/*Q3.Create a class time 1 which has three data members: hours, minute
and
second. Overload the ++ (post and pre) and -- (post and pre )operator
for the
class. Using friend and member function.
 Create a class which stores time1 in hh:mm format. Include all the
constructors.
The parameterized constructor should initialize the minute value to
zero, if it
is not provided.*/
//using friend function
#include <iostream>
using namespace std;
class time1
{
    int hr_025;
    int min_025;
    int sec;

public:
    time1()
    {
        hr_025 = 0;
        min_025 = 0;
        sec = 0;
    }
    void input(int t, int j, int k) // parameterized
    {
        hr_025 = t;
        min_025 = j;
        sec = k;
    }
    friend void operator++(time1 &a, int)
    {
        a.hr_025++;
        a.min_025++;
        a.sec++;
    }
    friend void operator++(time1 &a)
    {
        ++a.hr_025;
        ++a.min_025;
        ++a.sec;
    }
    friend void operator--(time1 &a, int)
    {
        a.hr_025--;
        a.min_025--;
        a.sec--;
    }
    friend void operator--(time1 &a)
    {
        --a.hr_025;
        --a.min_025;
        --a.sec;
    }

    void display()
    {
        cout << hr_025 << ":" << min_025 << ":" << sec << endl;
    }

} a[2], t;
int main()
{
    int hr, min, sec,c;

    cout << "enter  time\n";
    for (int i = 0; i < 1; i++)
    {
        cout << "enter" << i + 1 << " time in  hr min sec";
        cin >> hr >> min >> sec;

        a[i].input(hr, min, sec);
    }
    cout<<"press1 to post ++\n";
    cout<<"press2 to pre ++\n";
    cout<<"press3 to post --\n";
    cout<<"press4 to pre --\n";
    cin>>c;
    switch (c)
    {
    case 1:
        a[0].display();
        a[0]++;
        a[0].display();
        break;
    case 2:
        a[0].display();
        ++a[0];
        a[0].display();
        break;

    case 3:
        a[0].display();
        a[0]--;
        a[0].display();
        break;
    case 4:
        a[0].display();
        --a[0];
        a[0].display();
        break;

    default:
        break;
    }

    return 0;
}
```

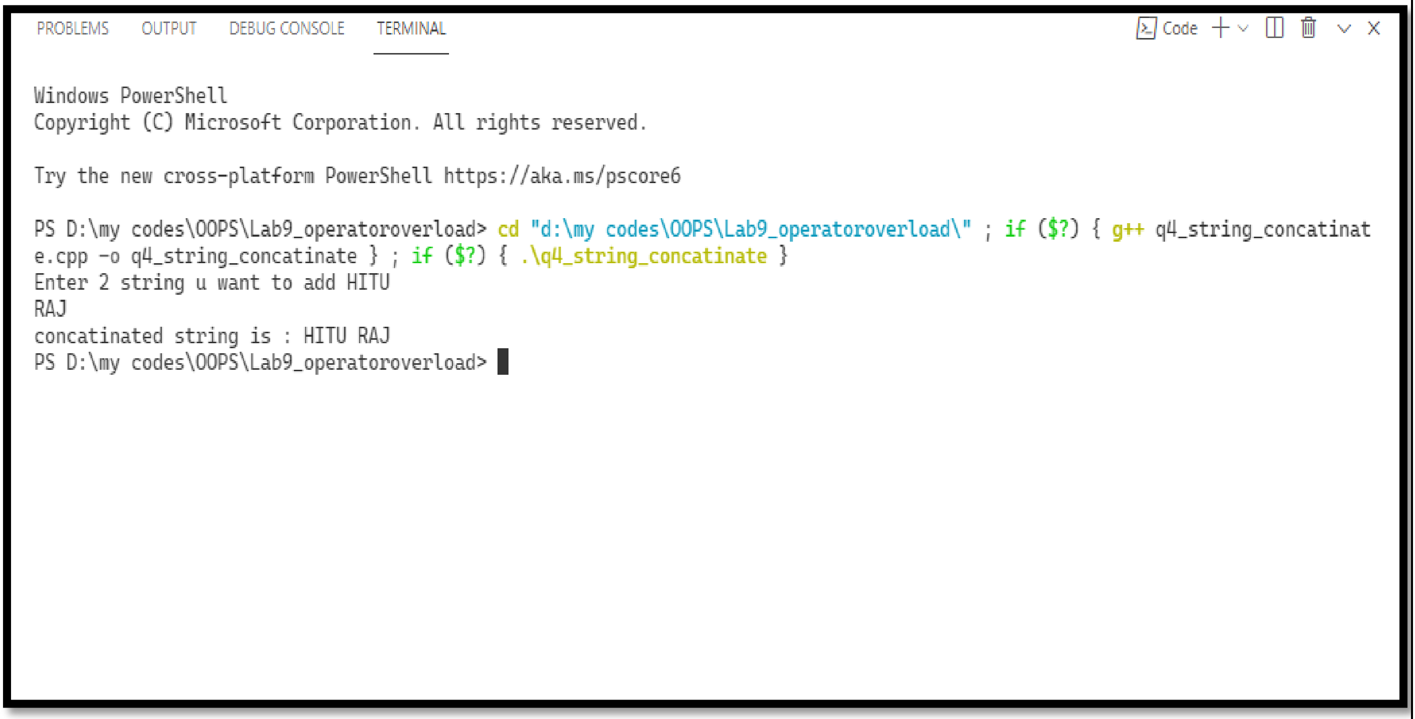
OUTPUT -3



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q3_time_pre.cpp -o q
3_time_pre } ; if ($?) { .\q3_time_pre }
enter  time
enter1 time in  hr min sec1 2 3
press1 to post ++
press2 to pre ++
press3 to post --
press4 to pre --
1
1:2:3
2:3:4
PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q3_time_pre.cpp -o q
3_time_pre } ; if ($?) { .\q3_time_pre }
enter  time
enter1 time in  hr min sec1 2 3
press1 to post ++
press2 to pre ++
press3 to post --
press4 to pre --
4
1:2:3
0:1:2
PS D:\my codes\OOPS\Lab9_operatoroverload> █
```

```
/*Q4Create a class which allocates the memory for a string through
dynamic
constructor. Overload the binary + to concatenate two strings and
display it.
Overload the assignment operator.*/
#include <string.h>
#include <iostream>
using namespace std;

class stringadd
{
    char *s;

public:
    stringadd(char *a)
    {
        s = new char[10];
        strcpy(s, a);
    }

    char* operator+(stringadd a)
    {
        return strcat(s, a.s);
    }
};
int main()
{
    char aa[10], be[10];
     char *res;//=new char[100];

    cout << "Enter 2 string u want to add ";
    gets(aa);
    gets(be);

    stringadd a(aa), b(be);
    res = a + b;
    cout<<"concatinated string is : "<<res;

    return 0;
}
```

OUTPUT -4

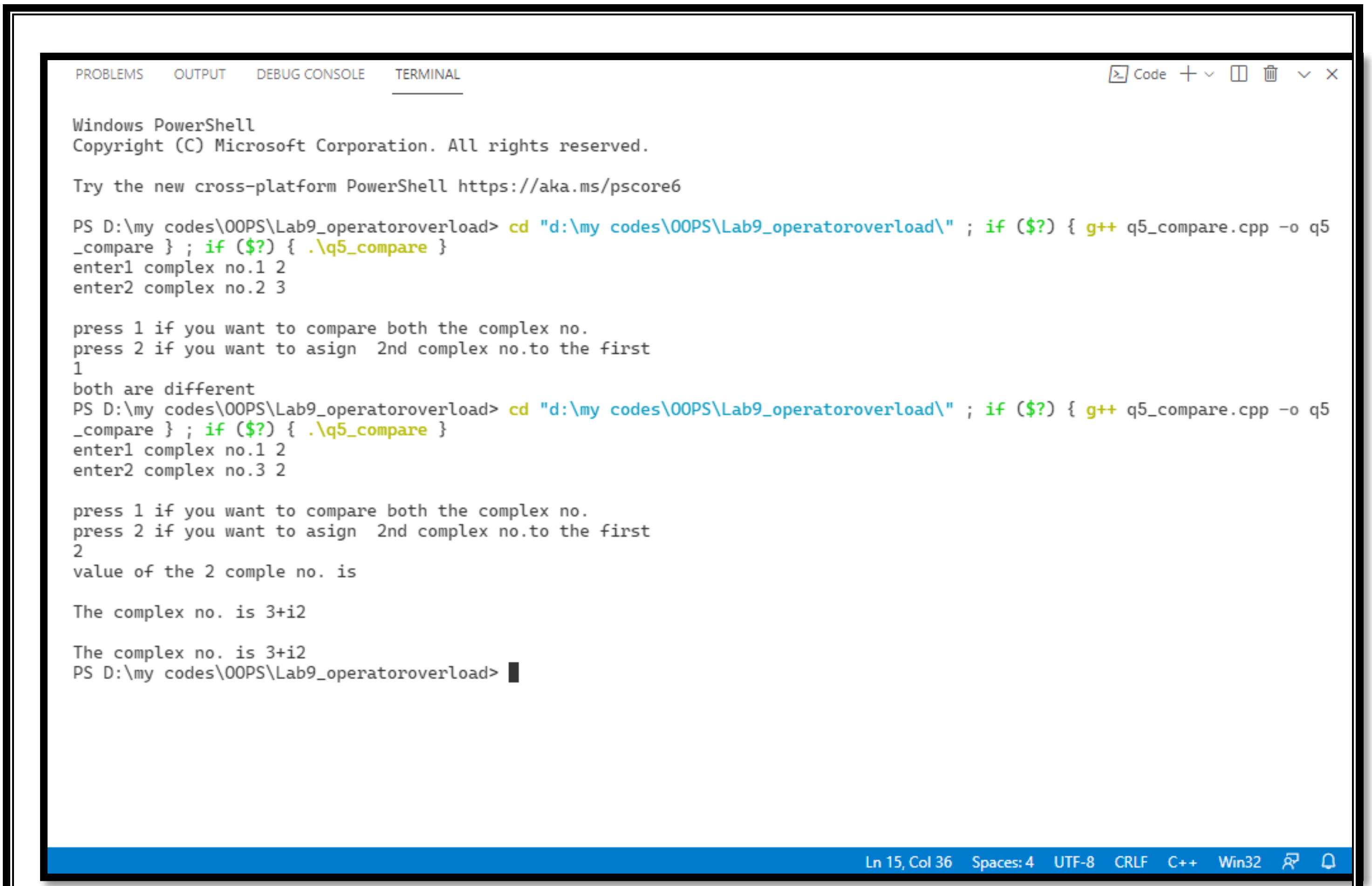


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q4_string_concatinat
e.cpp -o q4_string_concatinate } ; if ($?) { .\q4_string_concatinate }
Enter 2 string u want to add HITU
RAJ
concatinated string is : HITU RAJ
PS D:\my codes\OOPS\Lab9_operatoroverload> 
```

```
/*Q5 WAP to Overload the operator '==' to compare two objects of
complex class
```

```
and display whether they are equal or not. Overload the assignment
operator.*/

#include <iostream>
using namespace std;
class Complex
{
    int real_025;
    int img_025;
    ;

public:
    void input()
    {
        cin >> real_025 >> img_025;
        ;
    }
    bool operator==(Complex s2)
    {
        if (s2.real_025 == real_025 && img_025 == s2.img_025)
            return true;
        else
            return false;
    }
    Complex operator =(Complex s2)
    {
        real_025=s2.real_025;
        img_025=s2.img_025;
        return *this;
    }


    void display()
    {

        cout << "\nThe complex no. is " << real_025 << "+i" <<
img_025<<"\n";
        ;
    }
};
int main()
{
    int n_025, a_025;
    Complex s[2], c_025;
    for (int i = 0; i < 2; i++)
    {
        cout << "enter" << i + 1 << " complex no.";
        s[i].input();
    }
    cout
        << "\npress 1 if you want to compare both the complex no.";
      cout  << "\npress 2 if you want to asign  2nd complex no.to the
first\n";
    cin >> n_025;

    switch (n_025)
    {
    case 1:


        if (s[0] == s[1])
            cout << "both the complex no. are same\n";
        else
            cout << "both are different \n";

        break;
        case 2:


        s[0]=s[1];
        cout<<"value of the 2 comple no. is\n";
        s[0].display();
        s[1].display();
        break;

    default:
        break;
    }
    return 0;
}
```

OUTPUT -5

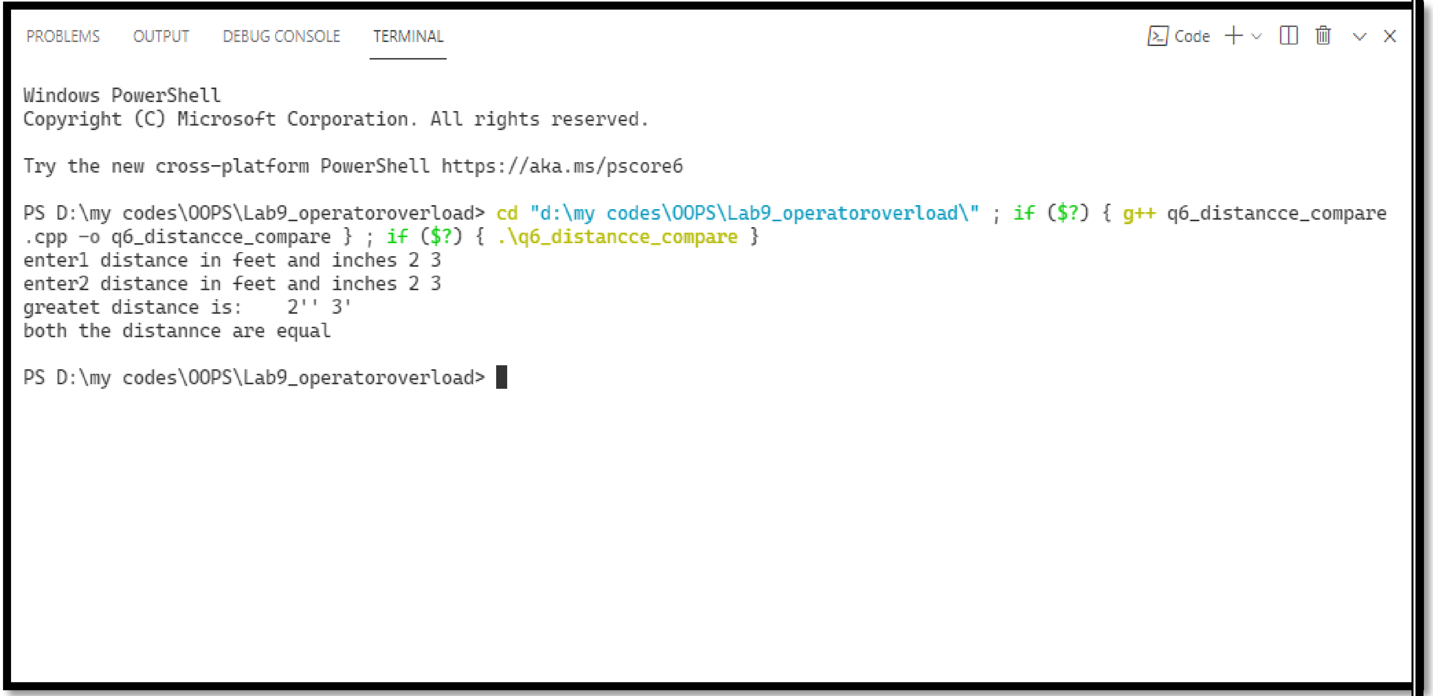


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q5_compare.cpp -o q5
_compare } ; if ($?) { .\q5_compare }
enter1 complex no.1 2
enter2 complex no.2 3

press 1 if you want to compare both the complex no.
press 2 if you want to asign  2nd complex no.to the first
1
both are different
PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q5_compare.cpp -o q5
_compare } ; if ($?) { .\q5_compare }
enter1 complex no.1 2
enter2 complex no.3 2

press 1 if you want to compare both the complex no.
press 2 if you want to asign  2nd complex no.to the first
2
value of the 2 comple no. is

The complex no. is 3+i2

The complex no. is 3+i2
PS D:\my codes\OOPS\Lab9_operatoroverload> 
```

```
/*Q6.WAP to add two objects of distance class. Overload the operator
‘>’ to
compare two objects and return the object with larger distance value
and
display it. Overload the ‘==’ operator to compare and display whether
two
given objects contain same distance value.*/

#include <bits/stdc++.h>
using namespace std;

class distance_025
{

    int feet_025;
    int inches_025;

public:
    void input()
    {

        cin >> feet_025 >> inches_025;
    }
```

```cpp
    void display()
    {
        cout << feet_025 << "'' " << inches_025 << "'\n";
    }
    bool operator>(distance_025);
    bool operator==(distance_025);
};

bool distance_025 ::operator>(distance_025 b)
{
    float ta, tb;
    ta = feet_025 + inches_025 / 12.0;
    tb = b.feet_025 + b.inches_025 / 12.0;

    return (ta > tb)
               ? true
               : false;
}
bool distance_025 ::operator==(distance_025 b)
{
    float ta, tb;
    ta = feet_025 + inches_025 / 12.0;
    tb = b.feet_025 + b.inches_025 / 12.0;

    return (ta == tb)
               ? true
               : false;
}
int main()
{
    distance_025 a[2];

    for (int i = 0; i < 2; i++)
    {
        cout << "enter" << i + 1 << " distance in feet and inches ";

        a[i].input();
    }
cout << "greatet distance is: \t";
    if (a[0] > a[1])
    {

        a[0].display();
    }
    else
        a[1].display();
    if (a[0] == a[1])

        cout << "both the distannce are equal \n ";
    else
        cout << "both are different\n";

    return 0;
}
```

OUTPUT -6



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\Lab9_operatoroverload> cd "d:\my codes\OOPS\Lab9_operatoroverload\" ; if ($?) { g++ q6_distancce_compare
.cpp -o q6_distancce_compare } ; if ($?) { .\q6_distancce_compare }
enter1 distance in feet and inches 2 3
enter2 distance in feet and inches 2 3
greatet distance is:    2'' 3'
both the distannce are equal

PS D:\my codes\OOPS\Lab9_operatoroverload>
```