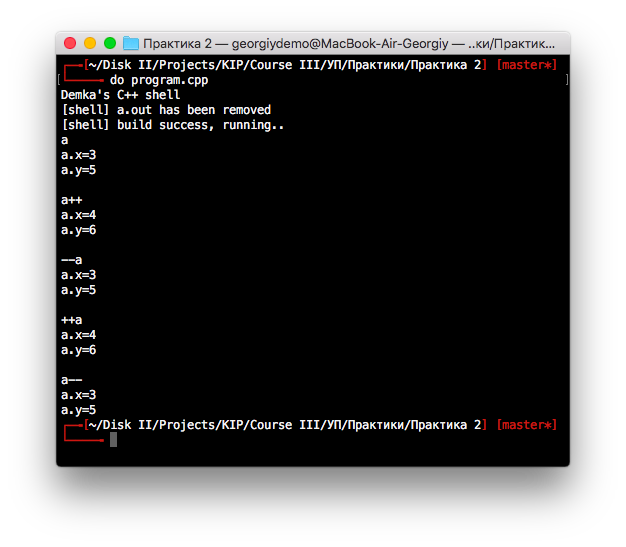
Условие:

Перегрузите оператор ++ (инкремента) и -- (декремента) относительно класса coord. Создайте их префиксные и постфиксные формы. Перегрузите оператор ++ (инкремента) и -- (декремента) относительно класса coord. Создайте их префиксные и постфиксные формы. Продемонстрируйте их работу.

Исходный код программы:#include **<iomanip>**#include **<iostream>**#include **<string>  
using namespace** std;  
  
**class** coord{  
  
 **public**:  
 **int** \_x, \_y;  
  
 coord(**int** x1 = 0, **int** y1 = 0)  
 {  
 \_x = x1;  
 \_y = y1;  
 };  
  
 coord& **operator**++(); *//Оператор префиксного инкремента.* coord **operator**++(**int**); *//Оператор постфиксного инкремента.* coord& **operator**--(); *//Оператор префиксного декремента.* coord **operator**--(**int**); *//Оператор постфиксного декремента.*};  
  
*//Оператор префиксного инкремента.*coord& coord::**operator**++()  
{  
 \_x++;  
 \_y++;  
 **return** \***this**;  
}  
  
*//Оператор постфиксного инкремента.*coord coord::**operator**++(**int**)  
{  
 coord temp = \***this**;  
 ++\***this**;  
 **return** temp;  
}  
  
*//Оператор префиксного декремента.*coord& coord::**operator**--()  
{  
 \_x--;  
 \_y--;  
 **return** \***this**;  
}  
  
*//Оператор постфиксного декремента.*coord coord::**operator**--(**int**)  
{  
 coord temp = \***this**;  
 --\***this**;  
 **return** temp;  
}  
  
**int** main(){  
  
 coord a(3, 5);  
 cout<<**"a\na.x="**<<a.\_x<<**"\na.y="**<<a.\_y<<**"\n\n"**;  
 a++;  
 cout<<**"a++\na.x="**<<a.\_x<<**"\na.y="**<<a.\_y<<**"\n\n"**;  
 --a;  
 cout<<**"--a\na.x="**<<a.\_x<<**"\na.y="**<<a.\_y<<**"\n\n"**;  
 ++a;  
 cout<<**"++a\na.x="**<<a.\_x<<**"\na.y="**<<a.\_y<<**"\n\n"**;  
 a--;  
 cout<<**"a--\na.x="**<<a.\_x<<**"\na.y="**<<a.\_y<<**"\n"**;  
  
**return** 0;  
}

Скриншоты программы:

a) Пример работы программы



б) Пример работы программы с другими исходными данными

