

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     x = 3;
6     y = x++;
7     printf("x = %d\n y = %d\n",x,y);
8     return 0;
9 }
10
11 /*
12     o/p would be x = 4 and y = 3
13
14     Explanation:
15
16     Initially x contains 3. When control reaches line 6, control goes to right
17     hand side of assignment operator since assignment operator has got lowest
18     precedence in the chart.
19
20     Here ++ operator is postfix to x and is a part of expression so behaviour
21     varies depending upon whether it is prefix or postfix...!
22
23     So here being a postfix increment operator, first, value of x is assigned
24     to y and then x is incremented i.e. after assignment. So y contains 3 and
25     x is incremented to 4
26 */
```

C-Free 4.0 - [C:\Cprog\prepost.c*]

File Edit Search View Project Build (Pro)Debug Tools Window Help

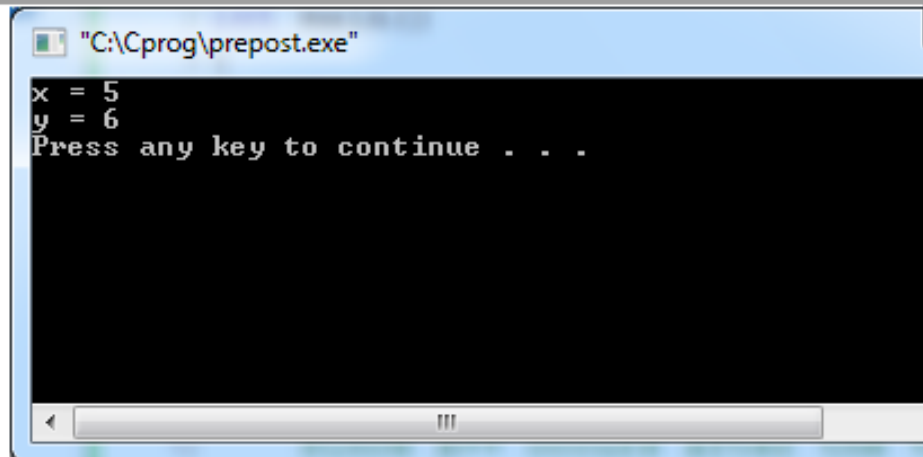
[R] mingw2.95

prepost.c*

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     x = 3;
6     y = ++x;
7     printf("x = %d\n y = %d\n",x,y);
8     return 0;
9 }
10
11 /*
12     o/p would be x = 4 and y = 4
13
14     Explanation:
15
16     Initially x contains 3. When control reaches line 6, control goes to right
17     hand side of assignment operator since assignment operator has got lowest
18     precedence in the chart.
19
20     Here ++ operator is prefix to x and is a part of expression so behaviour
21     varies depending upon whether it is prefix or postfix...!
22
23     So here, being a prefix increment operator,first,value of x is incremented
24     assigned then x is assigned to y. So y as well as x contains 4
25 */
```

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     x = 3;
6     y = ++x;
7     printf("x = %d\n y = %d\n",x,y++);
8     return 0;
9 }
10
11 /*
12 Here again o/p would be x = 4 and y = 4 though y in the printf statement
13 is incremented.
14
15 Explanation:
16
17 Initially x contains 3. When control reaches line 7, Here ++ operator
18 is postfix to y.
19
20 So first,value of y would be printed on the o/p window and then y is
21 incremented. Ultimately memory location of y would contain 5.
22 It would be confirmed if you try to print contents of y in the next
23 statement... (try it out)|
24 */
```

```
1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     x = 3;
6     y = x++ + x++;
7     printf("x = %d\n y = %d\n",x,y);
8     return 0;
9 }
10
11 /*
12     Since x++ occurs after the variable x, its value is first used to evaluate
13     the expression (line 6) and then x is incremented twice. Thus x + x would
14     result into 6 and then the result would be assigned to y. After this, the
15     first x would increment the value of x to 4, followed by second x++, which
16     would further increment x to 5
17 */
```



```

1 #include<stdio.h>
2 int main()
3 {
4     int x,y;
5     x = 3;
6     y = x++ + ++x;
7     printf("x = %d\n y = %d\n",x,y);
8     return 0;
9 }
10
11 /*

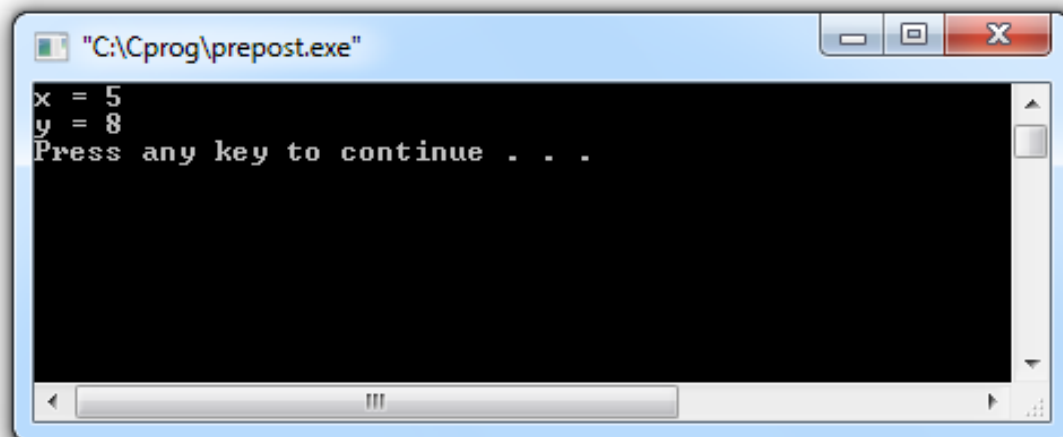
```

This one is bit tricky, follow it carefully. We might be led to believe that while evaluating z , what would be added is 3 and 4. But this is definitely wrong. This is because while calculating z , the very first operation that is performed is $++x$, which increments the value of x to 4. So by the time the addition $(x+x)$ is performed, x has already become 4. Thus the addition $4+4$ would be performed and not $3+4$. After this the result of the addition $4+4$ is assigned to the variable x . and then x is incremented to 5 (because of $x++$)

```

20 */

```



```

"C:\Cprog\prepost.exe"
x = 5
y = 8
Press any key to continue . . .

```