

Subject:

Year:

Month:

Date:

در تابع $mx1$ ، آدرس محل قرارگیری متغیرها را به عنوان آرگومان تابع در تعریفش در این فایل

این تابع از $\&$ استفاده کرده ایم بنابراین در $main$ به نام $\&$ خود متغیر را به تابع بدجیم می‌فرستیم

تابع $mx1$ آدرس محل قرارگیری متغیرها را گرفته است اما در خط ۴ انجام عملیات با مقادیر درون

آن آدرس‌ها جاری نمی‌شود و این را نیز بازی در می‌آوریم. در این حالت اگر تعریفی در خود متغیرها اعمال کنیم

در $main$ نیز قابل مشاهده خواهد بود اما اگر به جای آدرس متغیر، خود متغیر را به عنوان آرگومان تابع در تعریفش

این اتفاق نمی‌افتاد و هرگونه تعریف در متغیرها در $main$ اعمال نمی‌شد.

در تابع $mx2$ نیز آدرس متغیرها را به عنوان آرگومان می‌فرستیم اما این بار به صورت $pointer$.

در این صورت $\&a = \&c$ به این معنی است آدرس متغیر a در c ذخیره کرده ایم. بنابراین در آرگومان

تابع از $*int$ استفاده کرده ایم در نتیجه در $main$ باید آدرس متغیر را به تابع بدجیم. در اینجا نیز در

خط ۴ انجام عملیات با مقادیر درون آن آدرس‌ها جاری نمی‌شود $\leftarrow *m$ مقدار درون آن نشان می‌دهد

به صورتی $mx1$ خود متغیر را می‌گیرد ولی $mx2$ آدرس متغیر را و سپس با مقدار درون آن عملیات مورد نظر را انجام

می‌دهد. در هر دو تابع هر تعریفی در خود متغیرها در $main$ نیز اعمال نمی‌شود.

main.cpp

```
1  #include <iostream>
2
3  using namespace std;
4
5  int &max1 (int &m , int &n ) { return ( m > n ? m : n); }
6  int &max2 (int *m , int *n ) { return ( *m > *n ? *m : *n ); }
7
8  int main() {
9      int a = 5,b=3;
10     int *c = &a;
11     cout << "c = "<< c << endl;
12     cout << "max1 = " << max1 (a , b) << endl;
13     cout << "max2 = " << max2 (c , &b) << endl ;
14 }
15
```

input

c = 0x7ffdb1002298

max1 = 5

max2 = 5

...Program finished with exit code 0

Press ENTER to exit console.

main.cpp

```
1  #include <iostream>
2
3  using namespace std;
4
5  int &max1 (int &m , int &n ) {
6      m = 7;
7      return ( m > n ? m : n); }
8  int &max2 (int *m , int *n ) {
9      *n = 5 ;
10     return ( *m > *n ? *m : *n ); }
11
12 int main() {
13     int a = 5,b=3;
14     int *c = &a;
15     cout << "c = " << c << endl;
16     cout << "max1 = " << max1 (a , b) << endl;
17     cout << "a = " << a << endl ;
18     cout << "max2 = " << max2 (c , &b) << endl ;
19     cout << "b = " << b << endl ;
20 }
```



```
c = 0x7ffdc429c1d8
max1 = 7
a = 7
max2 = 7
b = 5
```