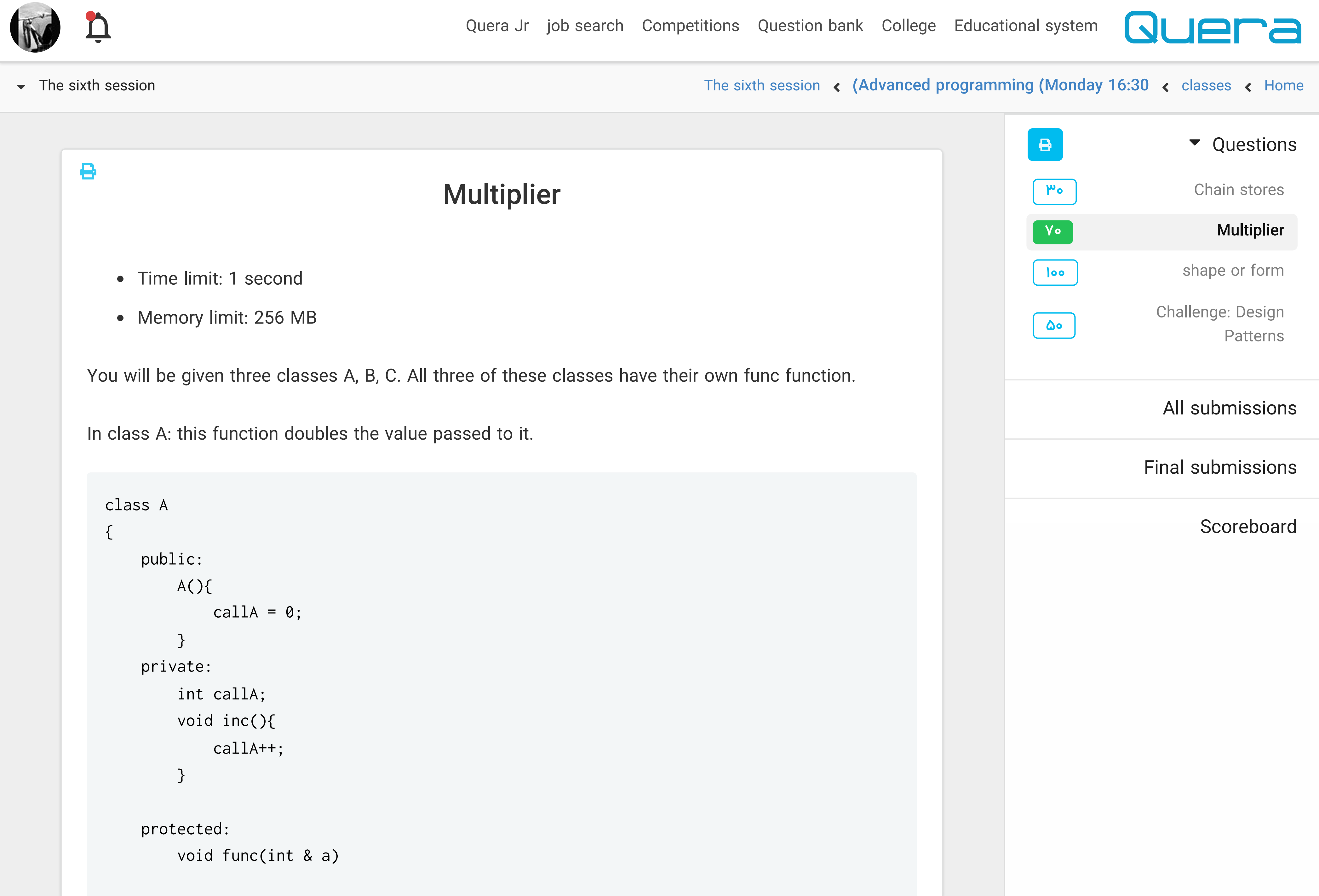

Multiplier

- Time limit: 1 second
- Memory limit: 256 MB

You will be given three classes A, B, C. All three of these classes have their own func function.

In class A: this function doubles the value passed to it.

```
class A
{
    public:
        A(){
            callA = 0;
        }
    private:
        int callA;
        void inc(){
            callA++;
        }

    protected:
        void func(int & a)
        {
            a = a * 2;
            inc();
        }
    public:
        int getA(){
            return callA;
        }
};
```

In class B: this function triples the value passed to it.

```
class B
{
    public:
        B(){
            callB = 0;
        }
    private:
        int callB;
        void inc(){
            callB++;
        }
    protected:
        void func(int & a)
        {
            a = a * 3;
            inc();
        }
    public:
        int getB(){
            return callB;
        }
};
```

In class C: this function multiplies the value passed to it by five times.

```
class C
{
    public:
        C(){
            callC = 0;
        }
    private:
        int callC;
        void inc(){
            callC++;
        }
    protected:
        void func(int & a)
        {
            a = a * 5;
            inc();
        }
    public:
        int getC(){
            return callC;
        }
};
```

You will be given class D in the following form:

```
class D
{

    int val;
    public:
        //Initially val is 1
         D()
         {
             val = 1;
         }

         //Implement this function
         void update_val(int new_val)
         {

         }
         //For Checking Purpose
         void check(int); //Do not delete this line.
};
```

Now you have to write the update_val function in such a way that it changes the value of val in class D to new_val. You should do this only by calling $ *func* $ functions in classes A, B, C. It is guaranteed that new_val is only a multiple of 2, 3, 5.

Input:

It contains only one line that takes the new_val value from the user

Output:

It will be output automatically by the following code, the sample format of which is available in the examples.

Example:

Input:

```
30
```

Output:

```
Value = 30
A's func called 1 times
B's func called 1 times
C's func called 1 times
```

Description:

At first, the value of val is one. Then the func function in class A is executed and the value of val is doubled. Then the func function in class B is executed and the value of val is tripled. Then the func function in C is executed and this time it is multiplied by 5, which becomes 30!

Your output code should be in the following form:

```
#include<iostream>
```

```
using namespace std;

class A
{
    public:
        A(){
            callA = 0;
        }
    private:
        int callA;
        void inc(){
            callA++;
        }

    protected:
        void func(int & a)
        {
            a = a * 2;
            inc();
        }
    public:
        int getA(){
            return callA;
        }
};

class B
{
    public:
        B(){
            callB = 0;
        }
    private:
        int callB;
        void inc(){
            callB++;
        }
    protected:
        void func(int & a)
        {
            a = a * 3;
            inc();
        }
    public:
        int getB(){
            return callB;
        }
};

class C
{
    public:
        C(){
            callC = 0;
        }
    private:
        int callC;
        void inc(){
            callC++;
        }
    protected:
        void func(int & a)
        {
            a = a * 5;
            inc();
        }
    public:
        int getC(){
            return callC;
        }
};

/*******************************/

class D
{

    int val;
    public:
        //Initially val is 1
         D()
         {
             val = 1;
         }

         //Implement this function
         void update_val(int new_val)
         {

         }
         //For Checking Purpose
         void check(int); //Do not delete this line.
};

/******************************/

void D::check(int new_val)
{
    update_val(new_val);
    cout << "Value = " << val << endl << "A's func called " << getA() << " times " << endl
}

int main()
{
    D d;
    int new_val;
    cin >> new_val;
    d.check(new_val);

}
```

POST AN ANSWER TO THIS QUESTION

.The training period is over

with Quera

Work with us
contact us
about us
Terms and Conditions
Sponsorship of competitions

events

Kodak
Scale up
Trainee exhibition
Tracey

Sources

Quora blog
Programmers' salary calculator
Statistics of the programming world
subscribe to newsletter

Products

Teaching programming
Recruitment ads
Programming questions
Competitions
classes
Employment platform
Quera Jr

Quera