Product of consecutive naturals X96568_en

Program the function:

```
// Pre: x>0
int product_of_consecutive(int x);
```

which, if x is the product of two consecutive naturals, returns the smaller of them. For instance, if $x = 42$, since $6 \cdot 7 = 42$, the function would return 6. If such a pair of naturals is not found, the function returns -1.
Use the following main program:

```
#include <iostream>
using namespace std;

int main() {
   int x;
   while (cin >> x) {
      int k = product_of_consecutive(x);
      if (k == -1) cout << x << " NO" << endl;
      else cout << x << " = " << k << "*" << k+1 << endl;
   }
}
```

IMPORTANT: The function `product_of_consecutive` must follow a binary search schema to solve this problem. Any solution not using binary search will be rendered INVALID.
HINT: Note that performing this binary search without a vector is analogous to use binary search to find the root of a function (seen in class). You also can see it as having an imaginary vector where `v[i]==i` (and since each position holds the same value of its index, the vector is not necessary).
For instance, if $x = 90$, we search for a number k between 1 and 90 such that $k * (k+1) = 90$.

```
(90+1)/2 = 45 --> 45*46 > 90, discard k>=45
(1+44)/2 = 22 --> 22*23 > 90, discard k>=22
(1+21)/2 = 11  -> 11*12 > 90, discard k>=11
(1+10)/2 = 5  --> 5*6 < 90, discard k<=5
(6+10)/2 = 8  --> 8*9 < 90, discard k<=8
(9+10)/2 = 9  --> 9*10 = 90
```

Exam score: 3.000000 **Automatic part:** 20.000000%

Input

A sequence of natural numbers, all of them greater than zero.

Output

For each natural x in the input, if x is the product of two consecutive naturals $x = k \cdot (k+1)$, the program writes both factors. Otherwise, the program writes "NO". Follow the format shown in the examples.

Sample input 1

```
42
91
90
812
332
81
156
```

Sample output 1

```
42 = 6*7
91 NO
90 = 9*10
812 = 28*29
332 NO
81 NO
156 = 12*13
```

Sample input 2

```
2
6
12
20
30
42
56
72
90
110
132
5
9
17
22
29
35
41
52
66
78
96
112
```

Sample output 2

```
2 = 1*2
6 = 2*3
12 = 3*4
20 = 4*5
30 = 5*6
42 = 6*7
56 = 7*8
72 = 8*9
90 = 9*10
110 = 10*11
132 = 11*12
5 NO
9 NO
17 NO
22 NO
29 NO
35 NO
41 NO
52 NO
66 NO
78 NO
96 NO
112 NO
```

Problem information

Author : Professors de PRO1
Generation : 2019-05-27 09:09:25