

## 1049.Brave Balloonists:

<http://acm.timus.ru/problem.aspx?space=1&num=1049>

### Обяснение:

Задачата е свързана с факторизиране на прости числа.

Задачата е следната, дадени са ни винаги на входа 11 числа

$$1 \leq a_i \leq 10000$$

Трябва да намерим колко са положителните делители на всяко число, и от там да изчислим  $a_1 * a_2 * a_3 * \dots * a_{10}$  като по този начин получаваме число N, търсим последната цифра на числото N. (това е отговора)

Така:

Генерираме прости числа до 10001, и факторизираме всяко едно от числото на прости делители като използваме един масив factors, в който ще пишем ++ на индекс-а когато намерим делител.

Формулата по която изчисляваме колко са делителите е следната:

$$n = p^a q^b r^c \dots$$

$$d(n) = (a+1)(b+1)(c+1) \dots$$

например за числото  $48 : 2^4 + 3^1$

съответно  $d(48) =$  вземаме степените  $(4+1) * (1+1) = 5 * 2 = 10$

накрая за всяко едно от числата, използваме горната формула :

$a_1 * a_2 * a_3 * \dots * a_{10}$  където  $a$  е броят на делителите на всяко число и след като го сметнем, печатим само последното число (отговора)

## Решение:

```
#include <string>
#include <iostream>
#include <vector>
#include <fstream>
#include <set>
#include <functional>
#include <map>
#include <algorithm>
#include <iterator>
#include <sstream>
using namespace std;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

int factors[10001];
int nums[11];
vector<int> p;
```

```

bool isprime(int p)
{
    for (int j = 2; j <= sqrt((double)(p)); j++)
    {
        if (p%j == 0)
        {
            return false;
        }
    }
    return true;
}

```

```

void genprimes(vector<int> &p)
{
    p.push_back(2);
    for (int i = 3; i < 10001; i+=2)
    {
        if (isprime(i)){
            p.push_back(i);
        }
    }
}

```

```

void factor(int z)
{
    for (int i = 0; i < p.size(); i++)
    {
        if (p[i] > z)
        {
            break;
        }
        else{
            while (z % p[i] == 0)
            {
                factors[p[i]]++;
                z = z/p[i];
            }
        }
    }
}

```

```

int main()
{
    genprimes(p);
    memset(factors, sizeof(factors), 0);

    for (int i = 0; i < 10; i++)
    {
        int d;
        scanf("%d", &d);
        factor(d);
    }
}

```

```
int sum = 1;
for (int i = 0; i < 10001; i++)
{
    int occurrence = 0;
    if (factors[i] != 0)
    {
        occurrence += factors[i] + 1;
        sum *= occurrence;
    }
}
stringstream ss;
ss << sum;
string str = ss.str();

printf("%c\n", str[str.length()-1]);

}
```