09-Simulation and Calculation

How to solve your problem by the computer?

There are two kinds of method:

- Simulation: Simulate human's every single step by the computer and get the result.
- Calculation: Get the calculation equation by logic reasoning and mathematical derivation, then use the computer to calculate the result avoiding tedious calculation work.

Sample Problem

A. New Year Candles

Vasily the Programmer loves romance, so this year he decided to illuminate his room with candles.

Vasily has a candles.When Vasily lights up a new candle, it first burns for an hour and then it goes out. Vasily is smart, so he can make b went out candles into a new candle. As a result, this new candle can be used like any other new candle.

Now Vasily wonders: for how many hours can his candles light up the room if he acts optimally well? Help him find this number.

Input

The single line contains two integers, a and b ($1 \le a \le 1000$; $2 \le b \le 1000$).

Output

Print a single integer — the number of hours Vasily can light up the room for.

Examples

input

Copy

```
4 2
```

output

Copy

```
7
```

input

Copy

```
6 3
```

output

Copy

```
8
```

Solution By simulation

Simulate to burn a new candles in every iteration

```
#include <iostream>

using namespace std;
int main()
{
  int a, b, ans = 0, c = 0;
  cin >> a >> b;
  while(a)     //Loop when there are still new candles
  {
      a--;    //In this loop, variable a means amount of new candles
      ans++;
      c++;     //variable c means amount of old candles
      if(c==b){
        a++;
        c-=b;
      }
  }
  cout << ans << endl;
}
```

Simulate to burn all new candles in every iteration

```
#include <iostream>
using namespace std;
int main()
{
  int a, b, ans = 0, c = 0;
  cin >> a >> b;
  while(a)      //Loop when there are still new candles
  {
```

```
    c += a;    //In this loop, variable a means amount of new candles
    ansn += a;
    a = c / b;  //variable c means amount of old candles
    c %= b;
  }
  cout << ans << endl;
}
```

Record different states and use different conditions to do branch and loop

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

int main() {
    int a, b, ans=0;
    scanf("%d%d", &a, &b);
    ans+=a;  //At first, burn all new candles,
    c = a;   //then get #a old candles
    while(c>=b) { //Loop when old candles are still more than #b, which means
that we can still make a new candle from old candles.
        ans+=c/b;
        c=c/b+c%b;  //In this loop, variable c means amount of old candles
    }
    printf("%d", ans);
    return 0;
}
```

Solution by Calculation

Every b old candles can make out 1 new candle, burn it 1 hour and left 1old candles, which means b-1 old candles can burn 1 hour.

We have #a new candles, which means we have a * b old candles in total.

However, for the last turn, if b-1 old candles remains, we cannot make out 1 new candle. Thus, we have to use a*b-1 to avoid this.

Thus, the final burning hour is (a * b — 1)/(b — 1) .

By the way, if we are permitted to borrow 1 old candles and to make 1 new candle, then to return it back after burning, we will can burn a*b/(b-1) hours.

```cpp
#include <iostream>
using namespace std;
int main()
{
  int a, b, n = 0, c = 0;
  cin >> a >> b;
  ans = (a * b – 1)/(b – 1)
  cout << ans << endl;
}
```