GOPAL'S DESTINATION

Problem **:** Gopal is allowed to make two types of moves: moves of length 1 meter, and moves of length B meters. Gopal is going to make exactly K moves **(all the moves along same direction)**. He would like to travel exactly D meters in those K moves. Can Gopal reach the destination? You are given D, K and B. Print **"Possible"** (quotes for clarity) if there is a combination of K jumps in same direction of total length equal to exactly D meters. Otherwise, print **"Impossible"**.

Difficulty : Easy

Explaination : Initially assume that all the "K" moves were done using 1meter jump, so the left over distance to cover is (D-K) which has to be adjusted in one of the K moves if it is possible. Adding a move of B metres in one of the K moves is nothing but removing a 1metre move and adding a B meter move i.e. adding (B-1) to the moves.

(And if B=1, then the only condition to check is if (D==k) or not)
i.e Let "m" be the no. Of 1 meter jump and "n" be the no. of B metres jump.
So,

$$m + (n*B) = D$$
$$m + n = K$$
$$n(B-1) = (D-K)$$

$n = (D-k)/(B-1)$, which is the number of B's to be added in the moves.
And if the expression, (total distance) – (B * no of B's) – (k-no of B's) = 0 , is satisfied then covering the D distance in K moves is possibke or else its Impossible.

Code (in C++) :

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
 long t;
 cin>>t;
 while(t--)
 {
  long d,k,b;
  cin>>d>>k>>b;
```

```
 if(b==1)
  {
    if(d==k)
    cout<<"Possible"<<endl;
    else
    cout<<"Impossible"<<endl;
  }
  else
  {
      long x = (d-k)/(b-1);
      if(x>=0 && (d-b*x-(k-x))==0 && x<=k)
        cout<<"Possible"<<endl;
       else
       cout<<"Impossible"<<endl;
  }
 }
}
```