C1-B

题目内容

题目描述

Zhoues 这个同学估计大家都不怎么认识，但是这不重要，重要的是你只需要知道他有一项特殊的能力——**暗中观察**。他经常通过这个能力轻松看透一些事物背后的算法原理。

虽然学校因为疫情封校，但是却校园内却盛行一款游戏：

- 该游戏中有小 A 和小 B 两个角色，初始给定两个正整数 n 和 k，小 A 和小 B 轮流对 n 减去 k 的任意非负整数次幂（即 $k^i, i \in \mathbb{N}$），但需要保证 n 始终非负，最终在自己回合将 n 变为 0 的一方获胜。

每次游戏开始都是小 A 先手，且小 A 小 B 两个人都会采取最优策略，那么每轮谁会赢呢？

Zhoues 暗中观察了几场这两款游戏的对局，已经看透了背后的算法奥秘，现在你只要告诉他游戏的初始情况，他就可以直接告诉你最后的赢家是谁！

输入格式

第一行一个正整数 t（$1 \le t \le 50$），表示数据组数。

对于每组数据，一行两个正整数 n, k（$1 \le n, k \le 10^3$），含义同题目描述。

输出格式

对于每组数据，输出一行：

- 若小 A 获胜，输出 `~A~`。
- 若小 B 获胜，输出 `~B~`。

输入样例

```
2
3 2
3 4
```

输出样例

```
~B~
~A~
```

样例解释

第一组数据中，小 A 有两种选择，减去 1 或者减去 2 ，但是无论小 A 选择哪种，小 B 都可以在他的回合将其变为 0 ，故小 B 获胜。

第二组数据中，由于 $n<k$ ，故小 A 和小 B 只能轮流减去 k 的 0 次幂（即 1），最后小 A 获胜。

Hint

可能大家第一次接触博弈的题目，这里简单说明一下博弈的思想：

- 如果有一个能**使对手进入必败状态**的方法，则**我方必胜。**
- 当我方拿到的条件**上手就直接输掉**，或者**我方的下一步怎么走**，**对方都必胜的话**，则**我方必败。**

题解思路

感觉杨振炜同学的题解太好了，几乎完全可以表达我的想法！！！

其实该题目的解题思路就是$Hint$中给出来的提示。

首先我们需要定义状态，在本题中状态就是当前 n 的值。

我们称一个状态是**必胜态**，当且仅当在此状态下，接下来需要行动的一方存在一种策略，使得对方接下来无论如何行动，己方最终都能获得胜利。我们称一个状态是**必败态**，当且仅当在此状态下，接下来需要行动的一方无论采取何种策略，都无法获得胜利。

容易发现，必胜态的一方采取最优策略后，对方一定是必败态；必败态的一方无论采取何种策略后，对方都一定是必胜态。因此一个状态是必胜态，当且仅当存在一个行动后的状态是必败态；一个状态是必败态，当且仅当所有行动后的状态都是必胜态。

那么具体的解题思路可以为如下：

我们考虑依次求出 $n=0,1,\ldots$ 时的状态时必胜态还是必败态。

不妨记录 $f(i)$，$f(i)=1$ 表示状态 i 是必胜态，$f(i)=0$ 表示状态 i 是必败态。当前状态为 n 时，下一轮对手可能得到的状态只有 $n-k^i\geq 0$，于是我们有

$$f(i)=\begin{cases}1 & ,\exists n-k^i\geq 0: f(n-k^i)=0\\ 0 & ,\text{otherwise}\end{cases}$$

边界条件是 $f(0)=0$，这是因为 $n=0$ 时接下来需要行动的一方已经输了。

单组数据的时间复杂度是 $\mathcal{O}(n\log k)$。

参考代码

```
#include <bits/stdc++.h>
#define maxn 1005
using namespace std;
```

```
int ans[maxn], vis[maxn];
int n, k;

int play(int x) {
    if(!x) return 0;
    if(vis[x]) return ans[x];
    vis[x] = 1;
    int res = 0;
    for(int i = 1; i <= x; i *= k) {
        res |= !(play(x - i));
    }
    ans[x] = res;
    return  res;
}

int main() {
    int t;
    scanf("%d",&t);
    while(t--) {
        scanf("%d%d",&n,&k);
        if(k == 1 || n < k) {
            puts(n%2==0 ? "~B~" : "~A~");
        } else {
            for(int i = 0; i <= n; i++)
            {
                vis[i] = 0, ans[i] = 0;
            }
            puts(play(n) ? "~A~" : "~B~");
        }
    }
    return 0;
}
```

Author: 周恩申