题目分析

这道题就是输入若干个数去求他们的最大值，那么我们很容易可以想到从头到尾遍历每个数，同时记录一下已经遍历过的数中最大的数，当遍历到新的数的时候，就和这个已经遍历过的最大值比较一下，如果比已有的最大值更大，那么就将最大值更新为新的值，同时用一个变量记录下来这个新的最大值对应的编号，否则继续遍历。

同时要注意，题目要求有多个最大值时要输出编号最大的，也就是靠后的数。因此如果遍历到了一个值和已有的最大值一样大，同样要将最大值的编号更新为这个靠后的数的编号。

示例代码

```
#include <stdio.h>
int main() {
    int n, t, maxV = -1, index; // maxV记录遍历过的数的最大值，用于比较. index 记录最大值的编号
    scanf("%d", &n);
    for (int i = 0; i < n; ++i) {
        scanf("%d", &t);
        if (t >= maxV) maxV = t, index = i + 1; // 这里是 >= 符号，因为等于的时候也应该更新index
    }
    printf("%d", index);

    return 0;
}
```

I 小P的乌龟爬水井 2

难度	考点
5	贪心、分类

题目分析

下面是 `C1 - H - 小P的乌龟爬水井` 的题解

由题，我们可以发现，当 $a \geq h$ 时，乌龟总能在第 1 天到井口，故可将此类情况单独分出。

之后，当 $a < h$ 时，可再分为两种情况。

1. 当 $a \leq b$ 时，由于乌龟上升距离不大于下降距离，故其每天开始时总位于井底。又由于其不能一次爬到井口，即 $a < h$ ，故其始终不能到达井口。

2. 当 $a > b$ 时，由于乌龟上升距离大于下降距离，故其每天的开始高度总比上一天高。故而，其总能到达井口。并且，由于夜晚的高度会下降，故乌龟一定在最后一天的白天就到达了终点。设第 ans 到达了井口，则一定有 $(ans-1)*(a-b)+a \geq h$ 其中 $(ans-1)*(a-b)$ 代表前 $ans-1$ 天行进的路程，而 a 代表最后一天行进的路程，二者相加应大于 h。变形可得 $ans \geq (h-a)/(a-b)+1$，即 $ans = \lceil (h-a)/(a-b) \rceil + 1$ ，直接计算即可。

以上为 `C1 - H` 的题解，下面将讲述如何将其他范围的 a 和 b 转换为只有自然数范围的 a 和 b。

1. $a < 0, b \geq 0$ 时：由于乌龟两次回落均在同一天，故若将其视为一次回落并不会影响最后结果。因而可将两次回落进行合并，并将向上爬距离视为 $0cm$。即令 $a_1 = 0, b_1 = b - a$，此时等价于乌龟白天向上爬 $a_1 cm$，晚上向下爬 $b_1 cm$，之后便可转换为 $C1H$ 的程序进行计算。
2. $a \geq 0, b < 0$ 时：同理，由于乌龟两次向上爬均在同一天，故若将其视为一次向上爬并不会影响最后结果。因而可将两次向上爬进行合并，并将回落距离视为 $0cm$。即令 $a_1 = a - b, b_1 = 0$，此时等价于乌龟白天向上爬 $a_1 cm$，晚上向下爬 $b_1 cm$，之后便可转换为 $C1H$ 的程序进行计算。
3. $a < 0, b < 0$ 时：有两点要提前说明：由于乌龟最低位于井底，不会位于更低的地方，故而第一天白天的回落没有意义，可视为从第一天晚上的上升开始计算；因为此时为晚上上升，白天下降，故而乌龟总会于夜晚到达井口。在当前状况下的一天中，回落总是先于上爬出现的。但是因为第一天白天没有意义，若将每一天的夜晚与后一天的白天当作一天看待，即总体向前平移半天，便会发现其又变成 $C1H$ 的先上升后下降了。并且，因为总是在夜晚到达，在时间平移后仍在同一天到达井口，只有白天与夜晚的差别，故而二者答案没有任何差别。因此，可令 $a_1 = -b, b_1 = -a$，此时等价于乌龟白天向上爬 $a_1 cm$，晚上向下爬 $b_1 cm$，之后便可转换为 $C1H$ 的程序进行计算。

所以，不论是哪一种情况，最后都能简单地化为都是正数的情况去解决，当全是负数的时候，相当于 $a_1 = -b$, $b_1 = -a$ 的情况；当 a 正 b 负时，相当于 $a_1 = a - b$, $b_1 = 0$ 的情况；当 a 负 b 正时，相当于 $a_1 = 0$, $b_1 = b - a$ 的情况。希望同学们能学会转化问题的能力，把遇到的新问题转化为自己已经解决的问题。

第二点，发现有不少同学写的时候有诸如这样的代码：`h + a - b` 或者 `(h + a + b) - a` ，实际上，参考 `C1 - I - 军乐团破冰` 的提示，以及 P1 上的例子（十亿加二十亿不等于三十亿，而是一个负数）；即 `int` 的表示范围是有上限的，大概是 21 亿（准确的值是 `2147483647`），所以前面两个计算会超过 `int` 的表示范围，正确的做法是用诸如 `(a - b) / (a - b) = 1` 和 `(a - a) = 0` 这样的方法合并，避免出现 `h + a - b` 或者 `(h + a + b) - a` 。

PS: 出题人将乌龟和蜗牛记混了……

示例代码

```
#include<stdio.h>
signed main()
{
    int t,a,b,h,ans,temp;//此题中int就足够了
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d%d%d",&h,&a,&b);
        //以下为新增代码
        if(a<0)
        {
            if(b<0)//a1=-b b1=-a
            {
                temp=a;
                a=-b;
                b=-temp;
            }
            else//a1=0 b1=b-a
            {
                b=b-a;
                a=0;
            }
        }
        else
        {
            if(b<0)//a1=a-b b1=0
            {
                a=a-b;
                b=0;
            }
        }
        //以上为新增代码
        if(a>=h)//a>=h时，第一天白天就可到顶，与b无关
        {
            printf("1\n");
        }
        else if(a<=b)//a<h且a<=b时，总无法到顶
        {
            printf("Impossible\n");
        }
        else//b<a<h时，可通过表达式计算出天数
        {
            ans=(h-b-1)/(a-b)+1;//a/b向上取整等价于(a+b-1)/b的向下取整
            printf("%d\n",ans);
        }
    }
```