

题意

维护一个数列，要求提供以下两种操作：1、 查询操作。语法：Q L 功能：查询当前数列中末尾L

个数中的最大的数，并输出这个数的值。限制：L不超过当前数列的长度。2、插入操作。语法：A n 功能：将n加

上 t ，其中 t 是最近一次查询操作的答案（如果还未执行过查询操作，则 $t=0$ ），并将所得结果对一个固定的常数 D 取

模，将所得答案插入到数列的末尾。限制：n是非负整数并且在长整范围内。注意：初始时数列是空的，没有一个数。

思路

序列只有增加和查询，没有删除，而且是往末尾添加的

举例：当前序列为 54 54 12 54 12 68 54 12 68 32 54 12 68 32 65
 54 12 68 32 65 66

→ → → →

→

查询结果为 54 54 12 68 68 68 68 68 68 32 68 68 68 65 65
 68 68 68 66 66 66

可以看出，如果新添加的一个数字比当前最后一个数字小，那么他不会前面的最小值造成影响。如果新添加的一个数字比当前最后一个数字大，那么可以看出，查询末尾两个的最大值是最后这个数字，而原来序列的最后一个数字不会在查询中出现，可以删掉，同样的，前面所有满足这个条件的，我们都可以删掉。上图涂色的数字则为每次添加数字后剩余的数。由于每次都只在末尾进行增加删除，我们可以用一个单调栈来维护。栈的元素为{ 涂色的值ans，下划线的部分开始的坐标x}。

如果用cin/cout会RE。。。。

代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long LL;
4  struct cre{
5      LL val;
6      int id;
7  };
8  cre st[300005];
9  int top=0;
10 int main()
11 {
12     int m;
13     LL D;
14     scanf("%d%lld",&m,&D);
15     char cmd;
16     LL num;
17     int n = 0;
18     LL ans = 0;
19     for(int i=1;i<=m;i++)
20     {
21         scanf(" %c",&cmd);
22         scanf("%lld",&num);
23         if(cmd == 'A')
24         {
25             n++;
26             cre tmp;
27             tmp.val = ans+num;
28             tmp.val %= D;
29             tmp.id = n;
30             while(top>0 && st[top].val <=tmp.val)
31             {
32                 tmp.id = st[top].id;
33                 top--;
34             }
35             st[++top] = tmp;
36         }
37         if(cmd == 'Q')
38         {
39             num = n-num+1;
40             int L = 1,R = top+1;
41             while(L<R)
42             {
43                 int mid = (L+R)/2;
44                 if(st[mid].id<=num)
45                     L = mid+1;

```

```
46         else
47             R = mid;
48     }
49     L--;
50     ans = st[L].val;
51     printf("%lld\n",ans);
52 }
53 }
54 }
55
```