题意

小明有n单位空闲时间，初始全为空，有3种操作
1. DS x 某屌丝约x连续时间，有则输出左区间，没有则拒绝
2. NS x 某女神约x连续时间，没有再判断鸽掉屌丝能不能约；有则输出左区间，没有则拒绝
3. STUDY!! x y 清空x 到 y 的计划

分析

本题的占用有两种情况，一种是女神的占用，一种是屌丝的占用，用一个线段树很难维护，所以干脆用两棵来维护，一棵tree维护屌丝和女神一起的占用。另一棵tree2维护只有女神的占用
每个结点维护三个信息：左端最长连续1(l)，右端最长连续1(r)，区间内最长连续1(tr)
清0，置1，都是常规操作
查询最早的x连续区间：
if 当前区间 l[o] >= x 返回该区间左端点
else if 左区间 tr[o<<1] >=x 查询左区间最早的x连续区间
else if r[o<<1] + l[o<<1|1] >=x 返回左区间最长右连续的端点
else 查询右区间最早的x连续区间

代码

```
#include<iostream>
#include<cstring>
#include<algorithm>
#include<cstdio>
#define For(i,a,b) for(int i=(a); i<=(b) ; i++)
#define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
#define Memset(a,b); memset((a),(b),sizeof((a)));
#define Cin(a); scanf("%d",&(a));
#define Cinc(a); scanf(" %c",&(a));
#define Cins(a); scanf("%s",(a));
#define Cout(a,b);  printf("%d",(a));printf(b);
#define Coutc(a,b);  printf("%c",(a));printf(b);
#define Couts(a,b);  printf("%s",(a));printf(b);
using namespace std;
typedef  long long LL;
typedef  unsigned long long ULL;
typedef  long double LDB;
```

```
18 inline int readint() {int x;cin>>x;return x;}
19 int  tree[500005],tree2[500005];
20 int l1[500005],r1[500005],l2[500005],r2[500005];
21 int lazy[500005],lazy2[500005];
22 int n,m;
23 char cmd[10];
24 int x,y;
25 int ans;
26 void pushdown1(int o,int l,int r)
27 {
28     if(lazy[o] == 1){
29         int M = (l+r)>>1;
30         lazy[o<<1] = lazy[o<<1|1] = 1;
31         tree[o<<1] = l1[o<<1] = r1[o<<1] = M-l+1;
32         tree[o<<1|1] = l1[o<<1|1] = r1[o<<1|1] = r-M;
33         lazy[o] = -1;
34     }
35     else{
36         tree[o<<1] = l1[o<<1] = r1[o<<1] = tree[o<<1|1] = l1[o<<1|1] =
   r1[o<<1|1] = 0;
37         lazy[o<<1] = lazy[o<<1|1] = 0;
38         lazy[o] = -1;
39     }
40 }
41 void pushdown2(int o,int l,int r)
42 {
43     if(lazy2[o] == 1){
44         int M = (l+r)>>1;
45         lazy2[o<<1] = lazy2[o<<1|1] = 1;
46         tree2[o<<1] = l2[o<<1] = r2[o<<1] = M-l+1;
47         tree2[o<<1|1] = l2[o<<1|1] = r2[o<<1|1] = r-M;
48         lazy2[o] = -1;
49     }
50     else{
51         tree2[o<<1] = l2[o<<1] = r2[o<<1] = tree2[o<<1|1] = l2[o<<1|1] =
   r2[o<<1|1] = 0;
52         lazy2[o<<1] = lazy2[o<<1|1] = 0;
53         lazy2[o] = -1;
54     }
55 }
56 void build(int o,int l,int r)
57 {
58     tree[o] = tree2[o] = l1[o] = l2[o] = r1[o] = r2[o] = r-l+1;
59     if(l==r){
60         return;
61     }
```

```
62      int M = (l+r)>>1;
63      build(o<<1,l,M);
64      build(o<<1|1,M+1,r);
65  }
66  void query1(int o,int l,int r,int d)
67  {
68
69      if(l1[o]>=d){
70          ans = l;
71          return ;
72      }
73      if(l >= r) return;
74      if(lazy[o] != -1) pushdown1(o,l,r);
75      int M = (l+r)>>1;
76      if(tree[o<<1]>=d) query1(o<<1,l,M,d);
77      else if(r1[o<<1]+l1[o<<1|1]>=d) {
78          ans  = M - r1[o<<1]+1;
79          return;
80      }
81      else query1(o<<1|1,M+1,r,d);
82  }
83  void query2(int o,int l,int r,int d)
84  {
85
86      if(l2[o]>=d){
87          ans = l;
88          return ;
89      }
90      if(l >= r) return;
91      if(lazy2[o] != -1) pushdown2(o,l,r);
92      int M = (l+r)>>1;
93      if(tree2[o<<1]>=d) query2(o<<1,l,M,d);
94      else if(r2[o<<1]+l2[o<<1|1]>=d) {
95          ans  = M - r2[o<<1]+1;
96          return;
97      }
98      else query2(o<<1|1,M+1,r,d);
99  }
100 void fill1(int o,int l,int r,int L,int R)
101 {
102     if(l>=L && r<=R){
103         lazy[o] = 0;
104         tree[o] = l1[o] = r1[o] = 0;
105         return;
106     }
107     if(lazy[o] != -1) pushdown1(o,l,r);
```

```
108     int M = (l+r)>>1;
109     if(M>=L) fill1(o<<1,l,M,L,R);
110     if(M+1<=R) fill1(o<<1|1,M+1,r,L,R);
111     l1[o] = l1[o<<1];
112     if(l1[o<<1] == M-l+1) l1[o]+=l1[o<<1|1];
113     r1[o] = r1[o<<1|1];
114     if(r1[o<<1|1] == r-M) r1[o]+=r1[o<<1];
115     tree[o] = max(tree[o<<1],tree[o<<1|1]);
116     tree[o] = max(tree[o],r1[o<<1]+l1[o<<1|1]);
117     return;
118 }
119 void fill2(int o,int l,int r,int L,int R)
120 {
121     if(l>=L && r<=R){
122         lazy2[o] = 0;
123         tree2[o] = l2[o] = r2[o] = 0;
124         return;
125     }
126     if(lazy2[o] != -1) pushdown2(o,l,r);
127     int M = (l+r)>>1;
128     if(M>=L) fill2(o<<1,l,M,L,R);
129     if(M+1<=R) fill2(o<<1|1,M+1,r,L,R);
130     l2[o] = l2[o<<1];
131     if(l2[o<<1] == M-l+1) l2[o]+=l2[o<<1|1];
132     r2[o] = r2[o<<1|1];
133     if(r2[o<<1|1] == r-M) r2[o]+=r2[o<<1];
134     tree2[o] = max(tree2[o<<1],tree2[o<<1|1]);
135     tree2[o] = max(tree2[o],r2[o<<1]+l2[o<<1|1]);
136     return;
137 }
138 void clear1(int o,int l,int r,int L,int R)
139 {
140     if(l>=L && r<=R){
141         lazy[o] = 1;
142         tree[o] = l1[o] = r1[o] = r-l+1;
143         return;
144     }
145     if(lazy[o]!=-1) pushdown1(o,l,r);
146     int M = (l+r)>>1;
147     if(M>=L) clear1(o<<1,l,M,L,R);
148     if(M+1<=R) clear1(o<<1|1,M+1,r,L,R);
149      l1[o] = l1[o<<1];
150     if(l1[o<<1] == M-l+1) l1[o]+=l1[o<<1|1];
151     r1[o] = r1[o<<1|1];
152     if(r1[o<<1|1] == r-M) r1[o]+=r1[o<<1];
153     tree[o] = max(tree[o<<1],tree[o<<1|1]);
```

```
154     tree[o] = max(tree[o],r1[o<<1]+l1[o<<1|1]);
155 }
156 void clear2(int o,int l,int r,int L,int R)
157 {
158     if(l>=L && r<=R){
159         lazy2[o] = 1;
160         tree2[o] = l2[o] = r2[o] = r-l+1;
161         return;
162     }
163     if(lazy2[o]!=-1) pushdown2(o,l,r);
164     int M = (l+r)>>1;
165     if(M>=L) clear2(o<<1,l,M,L,R);
166     if(M+1<=R) clear2(o<<1|1,M+1,r,L,R);
167     l2[o] = l2[o<<1];
168     if(l2[o<<1] == M-l+1) l2[o]+=l2[o<<1|1];
169     r2[o] = r2[o<<1|1];
170     if(r2[o<<1|1] == r-M) r2[o]+=r2[o<<1];
171     tree2[o] = max(tree2[o<<1],tree2[o<<1|1]);
172     tree2[o] = max(tree2[o],r2[o<<1]+l2[o<<1|1]);
173 }
174 int main()
175 {
176     int _;
177     Cin(_);
178     For(T,1,_)
179     {
180         cout<<"Case "<<T<<":"<<endl;
181         Cin(n);Cin(m);
182         Memset(lazy,-1);
183         Memset(lazy2,-1);
184         build(1,1,n);
185         while(m--)
186         {
187             Cins(cmd);
188             if(cmd[0] == 'D'){
189                 Cin(x);
190                 ans = -1;
191                 query1(1,1,n,x);
192                 if(ans == -1)
193                 puts("fly with yourself");
194                 else{
195                     Cout(ans,",let's fly\n");
196                     fill1(1,1,n,ans,ans+x-1);
197                 }
198             }
199             if(cmd[0] == 'N'){
```

```
                Cin(x);
                ans = -1;
                query1(1,1,n,x);
                if(ans == -1){
                    query2(1,1,n,x);
                    if(ans == -1) puts("wait for me");
                    else{
                        Cout(ans,",don't put my gezi\n");
                        fill1(1,1,n,ans,ans+x-1);
                        fill2(1,1,n,ans,ans+x-1);
                    }
                }
                else {
                    Cout(ans,",don't put my gezi\n");
                    fill1(1,1,n,ans,ans+x-1);
                    fill2(1,1,n,ans,ans+x-1);
                }
            }
            if(cmd[0] == 'S'){
                Cin(x);Cin(y);
                clear1(1,1,n,x,y);
                clear2(1,1,n,x,y);
                puts("I am the hope of chinese chengxuyuan!!") ;
            }
        }

    }
}
```