#include <cstdio>

#include <set>

#include <cstring>

using namespace std;

typedef long long LL;

struct P

{

LL i;

} p, q;

struct FH

{

LL h, sc;

} h;

LL d[300010], c[300010];

bool operator < (P a, P b)

{

return d[a.i] > d[b.i] || d[a.i] == d[b.i] && a.i < b.i;

}

bool operator < (FH a, FH b)

{

return a.h < b.h || a.h == b.h && a.sc < b.sc;

}

set<P> s;

set<FH> H;

LL PRIME = 1000000007;

LL pp[300010];

int main()

{

LL n, l, t, i;

char colour[10];

set<P>::iterator I;

scanf("%lld%lld%s", &n, &t, colour);

memset(c, -1, sizeof(c));

h.h = 0;

pp[0] = 1;

for (i = 1; i <= 3\*n; i++)

pp[i] = pp[i-1]\*PRIME;

for (i = n; i < (n<<1); i++)

{

scanf("%lld", d+i);

if (i == n)

c[i] = (colour[0] == 'b');

else

c[i] = c[i-1]^1;

p.i = i;

s.insert(p);

h.h += d[i]\*pp[i];

}

h.sc = (colour[0] == 'b');

H.insert(h);

for (i = 1; i <= t; i++)

{

p = \*s.begin();

s.erase(s.begin());

l = d[p.i];

if (l == 1)

{

printf("%lld\n", i-1);

return 0;

}

if (c[p.i] == 0)

{

p.i++;

I = s.find(p);

if (I != s.end())

s.erase(I);

p.i--;

h.h -= l\*pp[p.i]+d[p.i+1]\*pp[p.i+1];

d[p.i] = ((l+1)>>1);

d[p.i+1] += (l>>1);

c[p.i+1] = c[p.i]^1;

h.h += d[p.i]\*pp[p.i]+d[p.i+1]\*pp[p.i+1];

p.i++;

s.insert(p);

p.i--;

}

else

{

p.i--;

I = s.find(p);

if (I != s.end())

s.erase(I);

p.i++;

if (c[p.i-1] == -1)

h.sc ^= 1;

h.h -= l\*pp[p.i]+d[p.i-1]\*pp[p.i-1];

d[p.i-1] += (l>>1);

d[p.i] = ((l+1)>>1);

c[p.i-1] = c[p.i]^1;

h.h += d[p.i]\*pp[p.i]+d[p.i-1]\*pp[p.i-1];

p.i--;

s.insert(p);

p.i++;

}

s.insert(p);

if (H.find(h) != H.end())

{

printf("%lld\n", i);

return 0;

}

H.insert(h);

}

printf("%lld\n", t);

return 0;

}