#include <cstdio>

#include <queue>

using namespace std;

#define P (1993)

int n;

long long d[20010];

long long pow[20010];

priority\_queue<long long> Q;

void build()

{

int i;

pow[0] = 1;

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

pow[i] = pow[i-1]\*P;

}

int check(int l)

{

int c, p, q, i;

long long h = 0;

for (i = 0; i < l; i++)

h = h\*P+d[i];

Q.push(h);

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

{

h = (h-d[i-1]\*pow[l-1])\*P+d[i+l-1];

Q.push(h);

}

l = c = 1;

p = Q.top(); Q.pop();

while (Q.size())

{

q = Q.top(); Q.pop();

if (p == q)

c++;

else

{

if (l < c)

l = c;

c = 1;

p = q;

}

}

if (l < c)

l = c;

return l;

}

int main()

{

freopen("patterns.in", "r", stdin);

freopen("patterns.out", "w", stdout);

int k, p, q, l, i;

scanf("%d%d", &n, &k);

for (i = 0; i < n; i++)

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

p = 1;

q = n-k+1;

build();

while (q-p > 1)

{

l = (p+q)>>1;

if (check(l) >= k)

p = l;

else

q = l;

}

if (check(q) >= k)

printf("%d\n", q);

else

if (check(p) >= k)

printf("%d\n", p);

else

printf("0\n");

return 0;

}