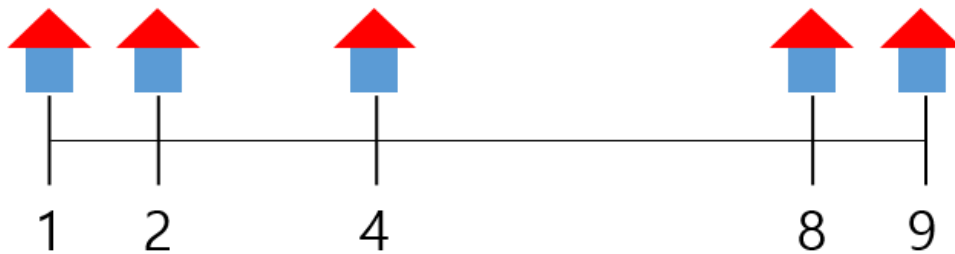


# 마을 재개발

집 간의 거리가 멀어 서로 왕래하기 힘들었던 인하 마을의 주민들은 마을을 재개발하여 집 간의 거리를 가깝게 하려고 한다. 인하 마을이 재개발되어 발전하는 것을 시기한 비룡 마을의 사람들이 재개발 설계도를 몰래 수정하여 인하 마을의 집들이 일렬로 지어지게 되었다. 공사가 끝난 뒤 이를 알아챈 인하 마을의 사람들은 별수없이 일렬로 나열된 집들 중에서 하나씩 선택하여 살기로 하였다. 새로 지은 집들을 구분하기 위해 각 집에는 번호가 정해져 있다. 가장 왼쪽에 있는 집의 번호는 항상 1번이고, 나머지 집들의 번호는 (1번 집과의 거리) + 1 번이다. 이웃 주민과의 거리가 가까우면 집 간의 소음이 걱정되어 인하 마을의 주민들은 가장 인접한 주민 간의 거리를 가능한 크게 하고 싶어 한다. 가장 인접한 주민 간의 거리가 가장 크게 집을 선택할 때, 가장 인접한 주민 간의 거리를 구해보자. 새로 지은 집의 개수는 항상 인하 마을의 주민의 수보다 크거나 같다.

다음 그림은 3명의 주민이 사는 인하 마을의 재개발 후의 지도이다. 3명의 주민이 각각 1번, 4번, 9번 집에 거주하면 가장 인접한 주민 간의 거리가 3으로, 가장 인접한 주민 간의 거리가 이보다 크게 할 수는 없다.



※ 프로그램의 실행 시간은 1 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

## 입력

첫 번째 줄에 인하 마을의 재개발 후 집의 수  $N$  ( $1 \leq N \leq 1,000,000$ )과 주민의 수  $M$  ( $2 \leq M \leq 10,000$ )이 주어진다.

두 번째 줄부터  $N$ 개의 줄에 걸쳐 집의 번호가 오름차순으로 주어진다.

1번 집은 항상 존재하며 집의 번호는 1,000,000,000보다 작은 양의 정수이다.

## 출력

가장 인접한 주민 간의 거리가 최대가 되도록 집을 선택할 때, 가장 인접한 주민 간의 거리를 출력한다.

예제 입출력

예제 입력	예제 출력
5 3 1 2 4 8 9	3