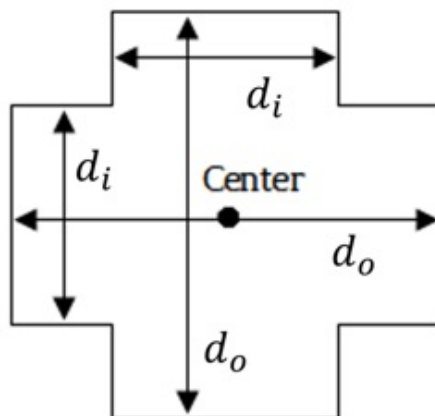


3-1. 십자가 놓기

십자가는 큰 정사각형의 네 꼭짓점과 한 꼭짓점이 겹치는 합동인 작은 네 개의 정사각형이 잘린 도형이다. 아래 그림은 안지름이 d_i 이고 바깥지름이 d_o ($d_i < d_o$)인 십자가를 의미한다.



N 개의 십자가 중에서 K 개를 적절히 골라 중심이 $(0, 0)$ 에 가고 변이 좌표축에 평행하도록 놓을 때, K 개가 모두 겹치는 영역의 최대 넓이를 구하는 프로그램을 작성하여라.

요구 사항

다음 함수를 구현해야 한다. 각 테스트 케이스에 대해 그레이더는 이 함수를 한 번 호출한다.

```
int64 SelectCross (int K, int[] I, int[] O)
```

- K : 선택하는 십자가의 수.
- I, O : 길이 N 의 정수 배열. i ($0 \leq i \leq N - 1$)번 십자가의 안지름은 $I[i]$ 이고 바깥지름은 $O[i]$ 이다.
- 이 함수는 십자가 K 개가 모두 겹치는 영역의 최대 넓이를 반환한다.
- 이 함수는 한 번 호출된다.

제한

- $2 \leq N \leq 200\,000$
- $1 \leq K \leq N$
- $1 \leq I[i] < O[i] \leq 1\,000\,000\,000$ ($0 \leq i \leq N - 1$)

부분문제

1. (8점) $K = 1$.
2. (55점) $K \leq 20$.
3. (37점) 추가 제약조건은 없다.

예제

다음 호출을 고려해보자.

```
SelectCross(3, [1, 2, 1, 1, 2], [2, 4, 3, 4, 3])
```

정답은 5이다.

샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks와 같은 IDE 툴을 사용한다면 `cross.cpp`, `cross.h`, `grader.cpp`를 한 프로젝트에 넣어서 컴파일하면 된다. 터미널에서 코드를 컴파일한다면 대회 페이지에 있는 컴파일 명령어를 이용하면 된다.

답안을 제출할 때에는 `cross.cpp`를 제출하면 된다.

Input format

- line 1: N K
- line 2: $I[0]$ $I[1]$ $\dots$ $I[N-1]$
- line 2: $O[0]$ $O[1]$ $\dots$ $O[N-1]$

Output format

샘플 그레이더는 `SelectCross` 함수의 반환값을 출력한다.