1-2. King of Chairs

Translator's note: In Korean, 'Uija' and 'chair' are both '의자,' and the problem title and statement contains a joke on this. To find who *King Uija* is, refer to this Wikipedia article: https://en.wikipedia.org/wiki/Uija_of_Baekje.

King Uija decided to have an interview to select maids of his palace. There are N candidates numbered 0 through $N-1$ in the order of arrival. For each i ($0 \le i \le N-1$), the weight of the candidate i is $W[i]$kg. King Uija knows the exact value of $W[i]$ (for all i) because he did some research on all the candidates. The weight of each candidate doesn't change forever.

As a chair mania, King Uija brought N colorful chairs in the waiting room. Chairs are also numbered 0 through $N-1$. For all j ($0 \le j \le N-1$), the *strength* of chair j is $C[j]$, which means in chair j only candidates whose weight is no more than $C[j]$kg can sit on. If some candidate who weighs strictly more than $C[j]$kg sits on chair j, the chair will be smashed, and King Uija will be furious since his precious chair is damaged. The chair can be occupied by at most one candidate.

Unfortunately for King Uija, he is an interviewer and has to be in the interview room. Thus there are only interviewees and his vassals in the waiting room. Since King Uija is a nice person, he ordered his vassals to maximize the number of interviewees who sit on his chairs. Note that King Uija knows the weights of all maids and the strengths of all chairs, and he computes the maximum number of candidates who can sit on his chairs based on this information.

To prevent chaos in the waiting room, whenever a candidate i arrives at the waiting room, the vassals will decide whether candidate i would sit on a particular chair j immediately. The candidate must follow the decision of the vassals, and stay still (either sitting on chair j or standing up) until all candidates arrive at the waiting room.

However, vassals initially have no information on the weights of candidates. Fortunately for them, there is a scale in the waiting room so that the vassals can measure the weight of the candidate who came into the waiting room correctly. However, there is no way for the vassals to know the weight of each candidate before they arrive at the waiting room. To resolve this situation, King Uija decided to send a telegram containing a non-negative integer to the vassals, so that they can have a rough idea about the weights of all maids.

You are a Yeonguijeong and have to devise a smart way to make as many interviewees as possible to sit on the king's chairs. After that, you should tell King Uija and the vassals what they should do to achieve the goal. Help them using a brand-new technology from the West, called 'computer.'

Implementation details

You have to submit two files.

The name of the first file is `king.cpp`. It represents the behavior of the king and should implement the following function. The file should include `king.h`.

```
int64 SendInfo(int[] W, int[] C)
```

- W: an array of length N. For each i ($0 \le i \le N-1$), $W[i]$ is the weight of candidate i.
- C: an array of length N. For each j ($0 \le j \le N-1$), $C[j]$ is the strength of chair j. C is sorted in non-decreasing order.
- This function is called exactly once at the beginning of the execution.
- This function should return a non-negative integer that the king is going to send to the vassals.

The name of the second file is `vassal.cpp`. It represents the behavior of the vassals and should implement the following functions. The file should include `vassal.h`.

```
void Init(int64 B, int[] C)
```

- B: the number that the king sent to the vassals. i.e. the return value of `SendInfo`.
- C: an array of length N. For each j ($0 \le j \le N-1$), $C[j]$ is the strength of chair j. C is sorted in non-decreasing order.
- This function is called exactly once at the beginning of the execution.

```
int Maid(int w)
```

- w: the weight of the candidate of maids who just arrived to the waiting room.
- If the vassals decide to make the candidate sit on chair j ($0 \le j \le N-1$), the function should return j. Otherwise, the function should return -1 instead.
- Note that each chair can be occupied by at most one candidate, so this function shouldn't return the same non-negative value two or more times.
- This function is called exactly N times after the function `Init` is called. The parameter w is given in the order $W[0], W[1], \cdots, W[N-1]$.
- The number of candidates who sit on a chair should be maximized.

If some of the above conditions are not satisfied, your program is judged as `Wrong Answer`. Otherwise, your program is judged as `Accepted` and your score is calculated by the maximum value of B (see Subtasks).

Grading Procedure

The grading is done in the following way. If your program is considered as Wrong Answer, it is terminated immediately.

1. The function `SendInfo` is called once.
2. The function `Init` is called once.
3. The function `Maid` is called N times. For the $(k+1)$-th call $(0 \leq k \leq N-1)$, the parameter `w` is $W[k]$. According to the return value of `Maid`, the grader counts the number of candidates who sits on the king's chairs, q.
4. If q is the same as the optimal value, your program is considered as correct.

Important Notices

- During the actual grading, these two programs are compiled and executed independently. They cannot share global variables.
- If runtime error or time/memory limit exceeded happens during the execution of the first program, the system doesn't execute the second program and gives that verdict. Your program is judged as `Wrong Answer` in this case.
- Both time and memory usage are measured by the sum of two processes.
- Your program should not use standard input and standard output. Your program should not communicate with other files by any methods. If you violate this, your program may be judged as `Wrong Answer`, but we cannot guarantee what would happen.

Example

Consider the following call in the first program made by `king.cpp`.

```
SendInfo([3, 6, 5], [3, 4, 5])
```

Suppose the return value of `SendInfo` is `365`.

Then, the following calls are made in the second program made by `vassal.cpp`.

```
Init(365, [3, 4, 5])
Maid(3)
Maid(6)
Maid(5)
```

For your program to be judged as `Accepted`, the return values of three `Maid` calls should be either `0, -1, 2` or `1, -1, 2`.

Constraints

- $1 \leq N \leq 100\,000$
- $1 \leq W[i] \leq 1\,000\,000$ (for all $0 \leq i \leq N-1$)
- $1 \leq C[j] \leq 1\,000\,000$ (for all $0 \leq j \leq N-1$)

Subtasks

1. (17 points) $1 \leq N \leq 10$
2. (38 points) W is sorted in non-decreasing order. In other words, $W[0] \leq W[1] \leq \cdots \leq W[N-1]$.
3. (45 points) No additional constraints.

Assume your program is judged as `Accepted`, and the king sent an integer B to the vassals. Then your score P for the test case, depending on its subtask number, is calculated as follows:

- Subtask 1. If $0 \leq B \leq 2^{60}$, $P = 17$. Otherwise, $P = 0$.
- Subtask 2. If $0 \leq B \leq 2^{20}$, $P = 38$. Otherwise, $P = 0$.
- Subtask 3.
  - If $B \geq 2^{60}$, $P = 0$.
  - If $2^{20} \leq B < 2^{60}$, $P = \left\lfloor 40 \times 0.97^{\lceil \log_2(B+1) \rceil - 20} \right\rfloor$.
    - Note that if $B = 2\,147\,483\,647$, $P = 28$.
  - If $X < B < 2^{20}$, $P = 40$.
  - If $B \leq X$, $P = 45$.
  - The value of X is hidden during the contest.

Note that your score for each subtask is the minimum of the scores for the test cases in the subtask.

Sample grader

You can download the sample grader package on the same page you downloaded the problem statement. (scroll down if you don't see the attachment)

If you use IDEs like Visual Studio, Eclipse or Code::Blocks, then import `king.cpp`, `king.h`, `vassal.cpp`, `vassal.h` and `grader.cpp` into one project and you will be able to compile all these files at once.

If you want to compile by yourself, refer to the following compilation command:

```
g++-7 -Wall -lm -static -DEVAL -o chairs -O2 grader.cpp king.cpp vassal.cpp -std=c++17
```

You should submit only `king.cpp` and `vassal.cpp`.

Input format

- line 1: N
- line 2: $W[0]\ W[1] \cdots W[N-1]$.
- line 3: $C[0]\ C[1] \cdots C[N-1]$.

Output format

If the vassals manage to make the candidates sit without errors, the sample grader prints `Correct` in the first line and q in the second line, with q the number of candidates who sit on the king's chairs. Note that if q is not the maximum possible number of candidates who can sit on the king's chairs, your program will be judged as `Wrong Answer`, although the sample grader prints `Correct`.

If your program is judged as `Wrong Answer`, the sample grader prints the error message in the first line.