2-3. Wine Tasting

Sandeul loves wine so much that he bought N types of wine numbered from 0 through $N-1$ and put all of them on display in his house. Wine i ($0 \leq i \leq N-1$) is the $R[i]$-th tastiest wine and contains some alcohol. Sandeul becomes drunk fast – so fast that he is not able to feel the essence of the taste of wine. Feeling sorry for Sandeul, you are going to hire a bartender so that Sandeul could savor the taste of wine.

The bartender can somehow blend the N wines, and make wine i (for each $0 \leq i \leq N-1$) contain exactly $A[i]$ milliliters of alcohol without changing the taste of wine. The bartender can choose the value of $A[i]$, but $A[i]$ has to be a positive integer due to technical difficulties.

The bartender can sort the N wines in the order of tastiness and could probably tell Sandeul which wine tastes the best. However, since you wanted to make Sandeul sort the wines in the order of tastiness on his own, you ordered the bartender not to tell Sandeul any information about the tastiness of wines. So, the bartender decided to tell Sandeul only the amount of alcohol contained in each wine, i.e., the sequence A.

Now, it's Sandeul's job to sort the wines in the order of tastiness. Sandeul can drink two wines i and j in a day and correctly know which wine is tastier. However, if Sandeul drinks more than K milliliters of alcohol in a day, he becomes drunk, thus being unable to determine which wine is tastier. Therefore, the amount of alcohol contained in the chosen two wines, i.e., $A[i]+A[j]$, should be no more than K milliliters. This implies that the bartender should make the amount of alcohol contained in each wine no more than K milliliters.

Sandeul wants to finish the job within 30 days. To help him, please write a program that implements the strategy of the bartender and Sandeul.

Implementation details

You have to submit two files.

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

```
int[] BlendWines(int K, int[] R)
```

- K: drinking capacity of Sandeul (in milliliters)

- R: an array of length N. For each $0 \leq i \leq N-1$, wine i is the $R[i]$-th tastiest wine.
- The function is called exactly once per test case.
- This function should return an array A of length N. For each $0 \leq i \leq N-1$), the value of $A[i]$ should be the amount of alcohol (in milliliters) contained in wine i. $1 \leq A[i] \leq K$ should hold.

The name of the second file is `taster.cpp`. It represents the behavior of Sandeul and should implement the following function. The file should include `taster.h`.

```
int[] SortWines(int K, int[] A)
```

- K: drinking capacity of Sandeul (in milliliters)
- A: an array of length N. For each $0 \leq i \leq N-1$, wine i contains $A[i]$ milliliters of alcohol. The array is the same as the array returned by `BlendWines`.
- The function is called exactly once per test case.
- This function should return an array r of length N. For each $0 \leq i \leq N-1$, the value of $r[i]$ should be equal to $R[i]$.

The function `SortWines` can call the following grader function:

```
int Compare(int P, int Q)
```

- P, Q: the indices of the wines that Sandeul will compare. $0 \leq P, Q \leq N-1$, $P \neq Q$, $A[P] + A[Q] \leq K$ should hold.
- If wine P is tastier than wine Q, this function returns -1. Otherwise, it returns 1.
- This function should be called at most 30 times per test case.

If some of the above conditions are not satisfied, your program is judged as `Wrong Answer`. Otherwise, your program is judged as `Accepted`.

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 `BlendWine` is called once.
2. The function `SortWines` is called once.
3. If everything is fine, your program is considered as `Accepted`.

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 `bartender.cpp`.

```
BlendWines(7, [1, 3, 2])
```

Suppose the return value of `BlendWines` is $[3, 4, 5]$.

Then, the following call is made in `taster.cpp`:

```
SortWines(7, [3, 4, 5])
```

Suppose `SortWines` made the following call to the grader function `Compare`:

```
Compare(0, 1)
```

The return value is -1. Notice that your program cannot call `Compare(0, 2)` because it makes Sandeul drunk.

For your program to be judged as `Accepted`, the return value of `SortWines` should be $[1, 3, 2]$.

Constraints

- $1 \leq N \leq 30$
- $7 \leq K \leq 30$
- $1 \leq R[i] \leq N$ (for all $0 \leq i \leq N-1$)
- $R[i] \neq R[j]$ (for all $0 \leq i < j \leq N-1$)

Subtasks

1. (100 points) No additional constraints.

Assume your program is judged as `Accepted` for all test cases where $K \geq C$ (if there are multiple such C, take the minimum one). The score P is determined as follows.

- If $7 < C \leq 30$, then $P = \lfloor 100 \times 0.87^{C-7} \rfloor$. Note that when $C = 30$, $P = 4$.
- If $C = 7$, then $P = 100$.

If there are no such $C \leq 30$ satisfying the condition, your score is 0.

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 `bartender.cpp`, `bartender.h`, `taster.cpp`, `taster.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 wine -O2 grader.cpp bartender.cpp t
aster.cpp -std=c++17
```

You should submit only `bartender.cpp` and `taster.cpp`.

Input format

- line 1: $N\ K$
- line 2: $R[0]\ R[1]\ \cdots\ R[N-1]$

Output format

If your program is judged as `Accepted`, the sample grader prints `Correct` in the first line.

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