



A Difficult(y) Choice (books)

Immortal glory goes to those who win a medal at BOI. As you are keen on being one of them, this is your way to go*: training, training, training!

As a first step in your training program, you decided to buy some computer science books. Luckily, your local book store has a special offer with a hefty discount when you buy exactly K books.

Now you are to select the K books to buy from the set of N computer science books (numbered 1 to N) offered in the book store. Your key selection factor is difficulty: Each book i has an individual (and fully objective) difficulty x_i , and the *total difficulty* of a set of books is the sum of their individual difficulties. You don't want the selected books to be too easy (then you wouldn't learn enough to win that precious medal) or too difficult (then you wouldn't understand them before the contest starts). To be precise, you want the total difficulty of the selected books to be at least A , but not more than $2A$.

Judging the actual difficulty of a book requires you to skim through it, but the store owner won't be happy if you read many books without buying them. She allows you to skim through at most S books. Fortunately, she also tells you that the books are sorted by increasing difficulty.

Write a program that assists you in deciding on which books to skim through, and in the end tells you which books to buy.

Communication

This is a communication task. You must implement the function `void solve(int N, int K, long long A, int S)` where N , K , A and S are as described above. The individual difficulties $x_1 < x_2 < \dots < x_N$ are initially kept hidden from your program. For each testcase, this function is called exactly once. In the function, you can use the following other functions provided by the grader:

- `long long skim(int i)` skims through the i -th book and returns its difficulty x_i ($1 \leq i \leq N$).
- `void answer(vector<int> v)` buys your selection of books. This function has to be called with $v = \{i_1, \dots, i_K\}$ ($1 \leq i_1, \dots, i_K \leq N$) where the i_j 's are pairwise distinct and satisfy $A \leq x_{i_1} + \dots + x_{i_K} \leq 2A$.
- `void impossible()` states that it is impossible to buy a set of K books with the desired difficulty.

If there exists a selection of K books with the desired difficulty, you must call `answer` exactly once. Otherwise, you must call `impossible` exactly once. Your program will be automatically terminated after a call to one of these two functions.

If any of your function calls does not match the above format, or if you call `skim` more than S times, your program will be immediately terminated and judged as **Not correct** for the respective testcase. You must not write anything to standard output, otherwise you may receive the verdict **Security violation!**.

If you use C++, you must include the file `books.h` in your source code. To test your program locally, you can link your program with `sample_grader.cpp` which can be found in the attachment for this task in CMS (see below for a description of the sample grader). The attachment also contains a sample implementation with additional explanations as `books_sample.cpp`.

If you use Python, you can find a sample implementation as `books_sample.py` in the attachment which also describes the interface for Python submissions.

* "Is it, really?," says a tiny voice in your head...



Constraints

We always have $K \leq N$, $3 \leq N$, $S \leq 10^5$, $1 \leq A$, $x_i \leq 10^{17}$, and $3 \leq K \leq 10$.

Subtask 1 (5 points). $S = N$, $170 \leq N \leq 1\,000$, $K = 3$

Subtask 2 (15 points). $S = N$, $N \geq 170$

Subtask 3 (10 points). $S \geq 170$, $x_{i+1} - x_i \leq A/K$ for all $1 \leq i \leq N - 1$

Subtask 4 (15 points). $S \geq 170$, $x_{i+1} - x_i \leq A$ for all $1 \leq i \leq N - 1$

Subtask 5 (15 points). $S \geq 170$

Subtask 6 (20 points). $S \geq 40$, $x_{i+1} - x_i \leq A$ for all $1 \leq i \leq N - 1$

Subtask 7 (20 points). $S \geq 40$

Sample Interaction

Consider a testcase with $N = 15$, $K = 3$, $A = 42$, and $S = 8$. At the beginning, the grader calls your function *solve* as *solve*(15, 3, 42, 8). Then, two possible interactions between your program and the grader could look as follows:

| Your program | Return value | Explanation |
|---|--------------|--|
| <i>skim</i> (1) <i>impossible</i> () | 1337 | The first (i.e., easiest) book has difficulty 1337. Thanks to your magical intuition, you have decided that there is no valid solution. The solution is correct and is accepted. |

| Your program | Return value | Explanation |
|-----------------------------|--------------|--|
| <i>skim</i> (1) | 7 | The first book has difficulty 7. |
| <i>skim</i> (15) | 21 | The last book has difficulty 21. Since the sequence of difficulties is strictly increasing, the sequence contains all integers from 7 to 21. Any valid set of values from the sequence with sum between 42 and 84 is acceptable. |
| <i>answer</i> ({11, 15, 7}) | | You answer with the book numbers 11, 15, 7, which correspond to difficulties 17, 21, 13. The solution is correct and is accepted. |

Grader

The sample grader expects on standard input two lines. The first line should contain the four integers N , K , A and S . The second line should contain a list of N integers, the sequence of difficulties $x_1 x_2 \dots x_N$ which has to be strictly increasing. Then, the grader writes to standard output a protocol of all grader functions called by your program. At the end, the grader writes one of the following messages to standard output:

Invalid input. The input to the grader via standard input was not of the above format.

Invalid skim. The function *skim* was called with invalid parameters.



Out of books to skim. The function *skim* was called more than S times.

Invalid answer. The function *answer* was called with invalid parameters.

Wrong answer. The function *answer* was called with a wrong selection of books.

No answer. The function *solve* terminated without calling *answer* or *impossible*.

Impossible (not checked): s book(s) skimmed. None of the above cases occurred, the function *skim* was called s times and the function *impossible* was called. It is not checked whether this is the correct answer.

Correct: s book(s) skimmed. None of the above cases occurred and the function *skim* was called s times.

In contrast, the grader which is used for the evaluation of your submission will only output **Not correct** (for any of the above errors) or **Correct**. Both the sample grader and the grader used for evaluation will terminate your program automatically whenever one of the above errors occurs or if your program calls *answer* or *impossible*.

Limits

Time: 1 s

Memory: 512 MiB