

米集積所 (Rice Hub)

田園地方に「米の道 (Rice Way)」として知られるまっすぐな道がある。この道に沿って R ヶ所の水田がある。各水田は 1 以上 L 以下の整数座標に位置する。これら R ヶ所の水田は座標の非減少列で表現される。形式的に書くと、 $0 \leq i < R$ なる任意の i に対して、水田 i は座標 $X[i]$ に位置する。 $1 \leq X[0] \leq \dots \leq X[R-1] \leq L$ と仮定して良い。

複数の水田が同じ座標となる場合があることに注意せよ。

収穫した米をなるべく多く貯蔵するために米集積所を1ヶ所建設する計画がある。水田と同様に、米集積所は 1 以上 L 以下の**整数座標に位置**しなければならない。水田が1つ、あるいは、2つ以上存在する座標を含めて、 1 以上 L 以下の整数座標であればどこにでも米集積所を建設できる。

各水田は収穫期ごとに**ちょうどトラック1台分**の米の収穫がある。集積所に米を輸送するため、市はトラック運転手を雇わなければならない。トラック1台分の米を集積所に輸送するのに運転手に支払う賃金は1単位距離あたり1パーツである。つまり、トラック1台分の米をある水田から米集積所に輸送するコストはそれらの座標の差に等しい。

残念なことに、今期の予算は厳しく、輸送に高々 B パーツしか支払えない。あなたの課題は、なるべく多くの米を集められるように米集積所の位置を決めるのを助けることである。

課題 (Your task)

次のパラメータを持つプロシージャー `besthub(R, L, X, B)` を実装せよ:

- R — 水田の数. 水田には 0 から $R-1$ の番号がふられている。
- L — 座標の最大値。
- X — 昇順に整列された整数の1次元配列. $0 \leq i < R$ なる各 i に対して、水田 i の座標は $X[i]$ である。
- B — 予算。

あなたが実装するプロシージャーは集積所の最適な位置を探し、予算内で集積所に輸送できる米は最大でトラック何台分かを返せ。

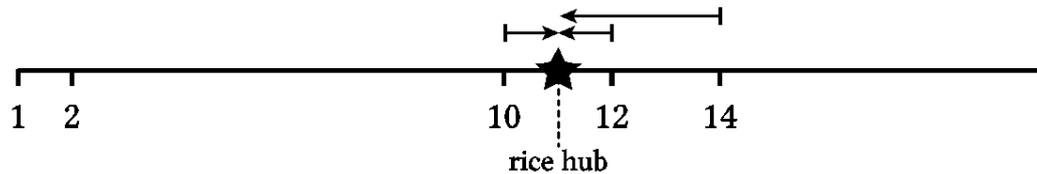
米の総輸送費は非常に大きくなる可能性があることに注意せよ。予算は 64-bit 整数で与えられる。解答では 64-bit 整数を使用することを勧める。C/C++ では `long long` 型を使用し、Pascal では `Int64` 型を使用せよ。

例 (Example)

例として, $R = 5$, $L = 20$, $B = 6$,

1
2
 $X = 10$
12
14

の場合を考える.



この場合は, 集積所の最適な位置は複数存在し, 10 以上 14 以下のどの位置に建設してもよい. 上の図はこれらの最適な位置の 1 つを示しており, 座標 10, 12, 14 の水田から米を輸送することができる. どの最適な位置に対しても, 総輸送費は 6 パーツを超えない. 明らかに, 3 ヶ所より多くの水田から米を集められる集積所の位置は存在しない. よって, この解は最適であり, **besthub** は 3 を返さなければならない.

小課題 (Subtasks)

小課題 1 (17 点)

- $1 \leq R \leq 100$
- $1 \leq L \leq 100$
- $0 \leq B \leq 10\,000$
- (この小課題に限っては) どの 2 つの水田も同じ座標ではない.

小課題 2 (25 点)

- $1 \leq R \leq 500$
- $1 \leq L \leq 10\,000$
- $0 \leq B \leq 1\,000\,000$

小課題 3 (26 点)

- $1 \leq R \leq 5\,000$
- $1 \leq L \leq 1\,000\,000$
- $0 \leq B \leq 2\,000\,000\,000$

小課題 4 (32 点)

- $1 \leq R \leq 100\,000$
- $1 \leq L \leq 1\,000\,000\,000$
- $0 \leq B \leq 2\,000\,000\,000\,000\,000$

実装の詳細 (Implementation details)

制限 (Limits)

- CPU 時間制限 (CPU time limit): 1 秒
 - メモリ制限 (Memory limit): 256 MB
- 注意:** スタックのサイズには決められた制限はない。スタックとして使用されたメモリは、メモリ総使用量に含まれる。

インターフェース (API) (Interface (API))

- 実装フォルダ (Implementation folder): `ricehub/`
 - 選手が実装するファイル: `ricehub.c` または `ricehub.cpp` または `ricehub.pas`
 - 提出ファイルのインターフェース (Contestant interface): `ricehub.h` または `ricehub.pas`
 - 採点プログラムのサンプル (Sample grader): `grader.c` または `grader.cpp` または `grader.pas`
 - 採点プログラムの入力のサンプル (Sample grader input): `grader.in.1`, `grader.in.2`, ...
- 注意:** 採点プログラムのサンプルは次の書式の入力を読み込む。
- 1 行目: **R** と **L** と **B**.
 - 2 行目から **R+1** 行目: 水田の位置; つまり, $0 \leq i < R$ に対し, **i+2** 行目には **X[i]** が書かれている。
 - **R+2** 行目: 期待される解。
- 採点プログラムの入力のサンプルに対して, 期待される出力: `grader.expect.1`, `grader.expect.2`, ...
この課題において, これらのファイルはいずれも文字列 “**Correct.**” のみを含むファイルである。