

1. 整数の配列 `data` から、値 `key` を探し出してその添字 (`index`: 配列の要素番号) を返す (戻す) メソッド (関数) を作成せよ。ただし、要素に値の重複はないものとし、もし `key` が `data` の中に含まれなかった場合には、`-1` を返すものとする。適当なクラス定義や `main` メソッドを補い、その動作を確認せよ。

```
public static int linearSearch(int key, int[] data)
{
    int ret = -1;

    return ret;
}
```

2. 上記の線形探索において、配列 `data` の中に値 `key` がないという最悪の場合、`data` の要素と `key` の比較は何回行われるか `n` を用いて表せ (つまりループを何回繰り返して `if` 文を何回実行するか)。さらに、配列のサイズ `n` が `k` 倍になると、繰り返しの回数は何倍になるか示せ。

3. 線形探索で、配列 `data` の中に値 `key` が必ず 1 個含まれる場合、さまざまな `data` と `key` の組合せで実行すると、探索 1 回あたり `data` の要素と `key` の比較は平均して何回行われるか、以下の手順で求めよ。

- (a) まず、平均の比較回数を直感的に予測し、`n` を用いて表してみよ。
- (b) `n` 個の中の 1 つの要素 `data[i]` に `key` が入っている確率 P_i を示せ。
- (c) `data[0]` から始めて `data[i]` までの `key` と要素の比較回数 L_i を示せ。
- (d) 平均の比較回数は以下の計算で求められる (すなわち $\sum P_i L_i$ である)。計算結果を `n` の式で表せ。

$$\begin{aligned}
 & \text{key が } data[0] \text{ にある確率 } P_0 \times \text{key が } data[0] \text{ にある場合の比較回数 } L_0 \\
 & \text{key が } data[1] \text{ にある確率 } P_1 \times \text{key が } data[1] \text{ にある場合の比較回数 } L_1 \\
 & \quad \cdot \cdot \cdot \\
 & +) \underline{\text{key が } data[n-1] \text{ にある確率 } P_{n-1} \times \text{key が } data[n-1] \text{ にある場合の比較回数 } L_{n-1}}
 \end{aligned}$$

4. 上記 2 の値を最悪計算量、上記 3 の値を平均計算量という。それぞれを O 記法で表せ。
5. 下記のように、Object クラスを用いれば、多態性 (ポリモーフィズム) によって任意のクラスに対応する線形探索が定義可能である。プログラムとして完成させ、Integer や String の探索を実行してみよ。

```
public static int linearSearch(Object key, Object[] data)
```