



古書 Ancient Books

德黑蘭市是伊朗國家圖書館的所在地。而當中最珍貴的寶庫位於一個狹長的大廳內，這個大廳內有 n 張桌子排成一行，從左到右順序編號為 0 到 $n - 1$ 。每一張桌子上都陳列著一本手抄的古書。這些古書是根據它的歷史年份排成順序的，但是這使客人們難以用書名來尋找出它們。所以，圖書館管理員決定按照字母順序來重新排列這些古書。

圖書館管理員 Aryan 被指派做這個工作。他做了一個長度為 n 的列表 p ，其中包含了由 0 到 $n - 1$ 中不同的整數。這個列表描述了按字母順序來重新排序這些古書的話所需要的改變：對於 $0 \leq i < n$ ，本來在桌子 i 上的古書最後應該要放到桌子 $p[i]$ 上。

Aryan 從桌子 s 開始把這些古書進行排序。他希望在完成工作之後走回到同一張桌子上。因為這些古書都是十分珍貴的，所以他每次只能搬動一本古書。當 Aryan 開始排序時他就會作出一個順序的操作。而每個操作只能是以下其中一個：

- 如果他手上沒有攜帶任何書而他所在的桌子上恰有一本書時，他可以拿起這本書。
- 如果他手上有攜帶了一本書而他所在的桌子上恰有一本書時，他可以把手上的書和桌子上的書交換。
- 如果他手上有攜帶了一本書而他所在的桌子上沒有書時，他可以把手上的書放到這個桌子上。
- 他可以走到任何一張桌子前。當他進行這個操作時，他手上可以攜帶著一本書。

對於 $0 \leq i, j \leq n - 1$ ，桌子 i 和桌子 j 之間的距離恰好是 $|j - i|$ 米。你的任務就幫助 Aryan 求出用這種方法來把書籍排好順序所需要行走的總距離的最小值。

編程詳細資料

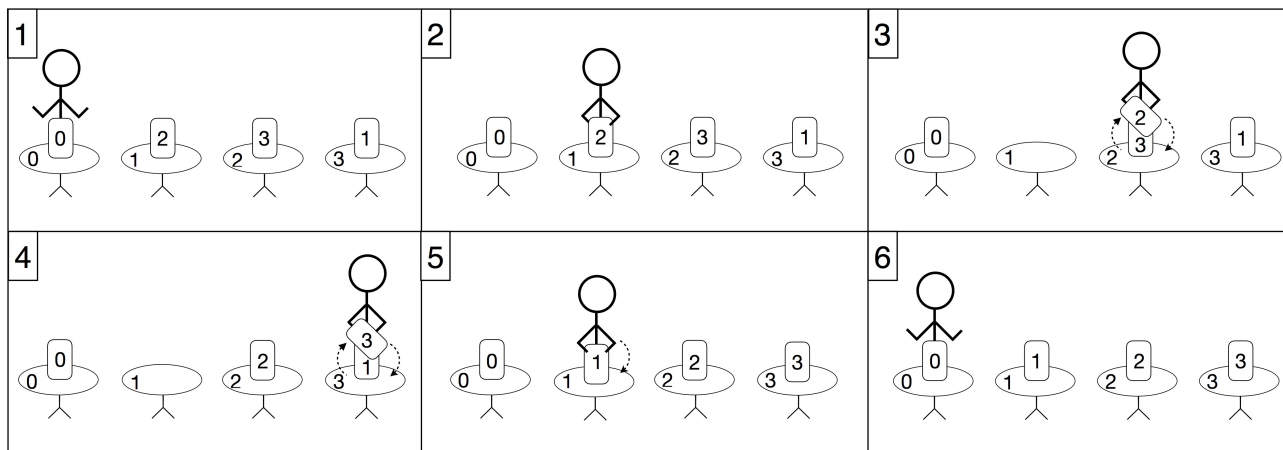
你應該編寫以下的子程序：

```
int64 minimum_walk(int[] p, int s)
```

- p 是一個長度為 n 的陣列。最初在桌子 i 上的古書應該要被 Aryan 拿到桌子 $p[i]$ 上。（其中 $0 \leq i < n$ ）
- s 就是 Aryan 開始時所在那張桌子的編號，而這也是他排序古書後應該停的位置。
- 這個子程序應該返回 Aryan 完成排序古書後所需要行走的總距離的最小值（以米來表示）。

樣例

```
minimum_walk([0, 2, 3, 1], 0)
```



在這個樣例中， $n = 4$ ，Aryan 在桌子 0 開始。他用以下的方式排序：

- 走到桌子 1 並且拿起桌上那本書。這本書應該要被放在桌子 2 上。
- 然後，他走到桌子 2 並且把他拿在手上的書和在桌子上的書進行交換。現在他手上那本新的書應該要放在桌子 3 上。
- 然後，他走到桌子 3 並且把他拿在手上的書和在桌子上的書進行交換。現在他手上那本新的書應該要放在桌子 1 上。
- 然後，他走到桌子 1 並且把他拿在手上的書放在桌子上。
- 最後，他回到桌子 0。

注意在桌子 0 上的那本書早已放到正確的地方，即桌子 0，所以 Aryan 並不需要拿起它。這個解答中他的總行走距離為 6 米。這是一個最優解；因此，子程序應該返回 6。

限制條件

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- 陣列 p 包含 n 個在 0 和 $n - 1$ 之間不同的整數。

子任務

1. (12 分) $n \leq 4$ 和 $s = 0$
2. (10 分) $n \leq 1000$ 和 $s = 0$
3. (28 分) $s = 0$
4. (20 分) $n \leq 1000$
5. (30 分) 沒有附加任何限制

樣例評測器

樣例評測器會讀取以下格式的輸入：

- 第 1 行: $n \ s$
- 第 2 行: $p[0] \ p[1] \ \dots \ p[n - 1]$

樣例評測器會打印一行輸出，其中包含minimum_walk的返回值。