

並再重新輸入) 之後, 依範例邏輯計算並印出各座位之號碼與效果值, 並輸出結果如下:

若以輸入 $n=12$ 為例:

```
Number of seats n in [5,20] = ? 12
Seat number: 11 9 7 5 3 1 2 4 6 8 10 12
Seat preference: 1 2 3 4 5 6 6 5 4 3 2 1
```

若以輸入 $n=3$ 再輸入 $n=15$ 為例:

```
Number of seats n in [5,20] = ? 3
!!WRONG!! Please input an integer in [5,20] = ? 15
Seat number: 15 13 11 9 7 5 3 1 2 4 6 8 10 12 14
Seat preference: 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
```

PART3 列印更新之後的座位號碼與喜好度:

請使用者輸入該排目前已被佔據 (即無法選取) 之座位號碼 (不需依大小排序), 以輸入 0 來終止此輸入。再度列印出座位號碼與喜好度, 但將上述已被佔據之座位的號碼及喜好度位置皆以「X」印出。接續上例如下: **[!!此部分不需要偵測輸入數值之正確性!!]**

```
Occupied seat no.: ? 9 1 3 4 8 10 0
Seat number: 15 13 11 X 7 5 X X 2 X 6 X X 12 14
Seat preference: 1 2 3 X 5 6 X X 7 X 5 X X 2 1
```

PART4 搜尋並列印出總喜好度最高的 K 個相鄰座位之號碼與總喜好度:

請使用者再輸入預備選擇的相鄰座位總數 K 之數值, 列印選位結果如下:

以上述 $K=4$ 為例, 因目前無法找到 4 個相鄰的剩餘座位, 則列印

```
How many adjacent seats to seek? 4
Sorry! No 4 adjacent seats!
```

否則, 若以 $K=3$ 為例, 則列印

```
How many adjacent seats to seek? 3
Best 3 adjacent seats: no. 15, 13, 11, with total preference 6.
```

同理, 若以 $K=2$ 為例, 則列印

```
How many adjacent seats to seek? 2
Best 2 adjacent seats: no. 7, 5, with total preference 11.
```

同理, 若以 $K=1$ 為例, 則列印

```
How many adjacent seats to seek? 1
Best 1 adjacent seats: no. 2, with total preference 7.
```

[!!此部分所選擇之座位不需要按特定大小順序列印出來!!]

解題建議

- 老師估計初學者必須花至少 1hr 才能將本題作好, 因為是第一個作業, 老師給大家兩週多來做。如果連這題你都必須花超過 5hr 才能作好, 代表之後你必須要更早開始寫作業。
- 本題分成 4 個 PART 處理, 前 2 個 PART 應該可以很快解決。
PART3 有很多種作法, 譬如你或可直接將被佔據的位子用另一個陣列儲存; 或者你可想辦法將被佔據的位子之 $S[]$ 或 $P[]$ 的值更新為 0, 再依該值是否為 0 來判斷是否列印「X」。不然, 你亦可考慮建構一個座位號碼的「反函數」陣列, 將其命名為 $I[j]$ 代表某座號 j 在 $S[]$ 的 index, 以 PART2 的 $n=12$ 例子而言, 假設 $S[] = \{11, 9, 7, 5, 3, 1, 2, 4,$

6,8,10,12}，則 $I[] = \{0,5,6,4,7,3,8,2,9,1,10,0,11\}$ 。如此一來，我們就知道座號為 4 的座位，其在 S[] 陣列的 index 是 $I[4]=7$ （亦即 $S[I[4]]=4$ ）。

PART4 稍微需要費一些心思，但相對於之後的 7 個作業而言，仍算是很簡單的。為了節省你 Debug 時間，不用每次 compile 完重新 run 都還要輸入那麼多數字有點煩，建議你在處理好 PART1、2、3 之後或可先將更新後的 S[]，P[] 寫死固定在程式中，直至你確認你搜尋過所有的「連續 K 個空位」之總喜好度，找出該連續座位的開頭位置，即可輕易完成題目要求，之後再將先前寫死的輸入資料改回原來的輸入方式。

此外，上課或實習課時會教授一個用 unix 指令 echo、<<與>>的方式將螢幕輸入的資料寫入檔案，再將該檔案餵回給執行檔，以省略繁瑣的螢幕手動輸入過程。

！特別注意事項！：

1. 這次作業老師特別提供一個 [sample 程式檔案](https://www.dropbox.com/s/9hr8c05t24sn92h/hw1_sample.cpp?dl=0) (https://www.dropbox.com/s/9hr8c05t24sn92h/hw1_sample.cpp?dl=0) 當起頭，幫忙大家早日熟悉相關要求，請多利用(以後不會再提供類似檔案，故請大家要好好觀摩一下 header 部分 comment 的寫法)。
2. 助教可能會輸入不同的 n 與 K 值來測試程式正確性，同學們應該多自行檢查看看。
3. 由於這是第一個作業，但必須用 cygwin 的 gcc 或 g++ compile 成功及 run，老師預期大部分同學可能需要花一至兩個小時以上的時間來熟悉相關軟體及作法，故請大家務必早點開始練習。

*作業繳交應注意事項

1. 作業需繳交電子檔以及書面(列印程式檔及程式結果)
2. 電子檔請於作業繳交截止時間以前上傳至 <http://moodle.ncku.edu.tw>
 - 2.1 請同學先建立一個資料夾，資料夾名稱為“學號_hw1”，例如學號為 hxxxx，則資料夾名稱則為 hxxxx_hw1
 - 2.2 將程式檔案名稱存為“hw1.cpp”，並將此程式檔案存於上述設立之學號_hw1 資料夾中
 - 2.3 最後將整個學號_hw1 資料夾壓縮成 zip 檔(學號_hw1.zip)，再上傳至 moodle 系統

(！注意！：請勿將 cpp 檔 copy/paste 至 word 檔而上傳之)
3. 書面作業請於 **2018/03/13 上課 5 分鐘內(09:15 前)**繳交至講台，其中需要註明程式是否能被編譯與執行、撰寫人、程式之目的、如何編譯及執行等資料(詳見 <https://www.dropbox.com/s/pvac59tfefggokd/programming.html?dl=0>)。