

## Handwriting

- 1.(35%) What would be the value of queues Q1 and Q2, and stack S after the following algorithm segment:

```
1 S = createStack
2 Q1 = createQueue
3 Q2 = createQueue
4 enqueue (Q1, 5)
5 enqueue (Q1, 6)
6 enqueue (Q1, 9)
7 enqueue (Q1, 0)
8 enqueue (Q1, 7)
9 enqueue (Q1, 5)
10 enqueue (Q1, 0)
11 enqueue (Q1, 2)
12 enqueue (Q1, 6)
13 loop (not emptyQueue (Q1))
    1 dequeue (Q1, x)
    2 if (x == 0)
        1 z = 0
        2 loop (not emptyStack (S))
            1 popStack (S, y)
            2 z = z + y
        3 end loop
        4 enqueue (Q2, z)
    3 else
        1 pushStack (S, x)
    4 end if
14 end loop
```

- 2.(35%) Imagine that the contents of queue Q1 and queue Q2 are as shown. What would be the contents of Q3 after the following code is executed? The queue contents are shown front (left) to rear (right).

Q1: 42 30 41 31 19 20 25 14 10 11 12 15

Q2: 4 5 4 10 13

```
1 Q3 = createQueue
2 count = 0
3 loop (not empty Q1 and not empty Q2)
  1 count = count + 1
  2 dequeue (Q1, x)
  3 dequeue (Q2, y)
  4 if (y equal count)
    1 enqueue (Q3, x)
  5 end if
1 end loop
```

- 3.(30%) Write an algorithm called `queueToStack` that creates a stack from a queue. At the end of the algorithm, the queue should be unchanged; the front of the queue should be the top of the stack, and the rear of the queue should be the base of the stack.

## Programming

- 4.(50%) One way to evaluate a prefix expression is to use a queue. To evaluate the expression, scan it repeatedly until you know the final expression value. In each scan read the tokens and store them in a queue. In each scan replace an operator that is followed by two operands with their calculated values. For example, the following expression is a prefix expression that is evaluated to 159:

- + \* 9 + 2 8 \* + 4 8 6 3

We scan the expression and store it in a queue. During the scan when an operator is followed by two operands, such as + 2 8, we put the result, 10, in the queue.

After the first scan, we have

- + \* 9 10 \* 12 6 3

After the second scan, we have

- + 90 72 3

After the third scan, we have

- 162 3

After the fourth scan, we have

159

Write a C program to evaluate a prefix expression.

- 請自行實作Queue, 不能使用Standard Library的容器(Container)。
- 實作之Queue應符合FIFO(First in First out)的使用原則。
- input/output皆使用標準串流(stdin/stdout), 如果沒有特殊情況, 就是指使用scanf/cin跟printf/cout,

以下為規範說明：

- 首先輸入一個正整數 $n$ ( $n \leq 100$ ), 代表接下來會有 $n$ 組測資。
- 每一組測資包含一個正整數 $m$ ( $3 \leq m \leq 200$ )跟一串prefix expression S,  $m$ 代表S由多少個token(運算符+數字)所組成。
- 運算符為基本四則運算, 用+ - \* /表示
- 數字則為絕對值小於1000之整數。每個token之間用一個空格隔開。
- S依題目敘述計算完畢後, 將結果output出來並換行, 繼續input下一組測資。
- $n$ 組測資的結果輸出完畢後, 結束程式即可。
- 單次計算若涉及小數, 請將結果的小數位無條件捨去後再進行後面的操作。  
(e.g: 1. // 30 4 3 => (30 / 4) / 3 => 7 / 3 => 2  
2. / -5 2 => -5 / 2 -> -2)
- 計算過程跟結果皆不會出現超過 $\pm 10^7$ 外的數值。
- 不會出現無法計算的S(e.g: + + 2 2 or / 1 0)。
- 請勿輸出其他無關文字(e.g “Please enter:”, “Your Answer:”, “End!!”)

範例：

Input

```
3
13
- + * 9 + 2 8 * + 4 8 6 3
3
- 1 3
7
- + -8 / -15 4 2
```

Output

```
159
-2
-13

```

5.(50%) 可以用STL, Time Limit : 10 sec

李哥是一名電競選手, 目前正在備戰冠軍賽, 但還是會有一堆粉絲電話打來找他。

假設今天會有  $N$  通電話, 對於  $i$  筆電話, 有這樣的資訊:

電話編號  $N_i$ , 電話抵達時間  $T_i(\text{sec})$ , 預計講電話時間  $L_i$ ,

若沒有delay, 則李哥會在  $[T_i, T_i + L_i)$  講電話 (一邊練習)。

如果電話打來的時候, 李哥還在跟之前的人對話, 打入的電話會等到他講完。

請你幫忙計算對於第  $i$  通電話,  $N_i$  開始跟李哥講電話的時間、李哥接到電話的瞬間總共有多少人在等待 (不包含  $N_i$  跟正講完電話的人)。以及所有人等待的時間總和。

(阿飛加油)

輸入說明:

此題為單筆輸出輸入, 輸出完畢結束程式即可。

首先會有  $N$  代表接下來會有  $N$  通電話

對於每通電話  $N_i$  都有兩個整數, 分別為撥入的秒數  $T_i$ 、預計通話的時間  $L_i$

$$0 \leq N \leq 5 * 10^5$$

$$0 < T_i \leq 10^6, \forall T_{i-1} < T_i$$

$$0 < L_i \leq 10^3$$

$$\text{保證 } \max(\text{QueueSize}) < 10^6$$

輸出說明:

總共要輸出  $N + 1$  行

前  $N$  行: 對於電話  $N_i$ , 李哥接到電話的瞬間、等待列隊的長度。

兩個數據請用空白隔開, 行尾直接換行即可

第  $N + 1$  行輸所有人等待的時間的總和

- 請勿輸出其他無關文字(e.g “Please enter:”, “Your Answer:”, “End!!”)

範例輸入:	範例輸出:
4	2 0
2 3	7 0
7 6	13 1
8 3	16 2
9 3	12

範例解釋:

第8 sec撥入時正在等待第7sec撥入的電話 (1)

第9 sec撥入時正在等待第7sec, 第8sec撥入的電話 (2)

而總共等待時間為  $(2-2) + (7-7) + (13-8) + (16-9) = 12$

## Submission - **Deadline: 2022/11/18 13:20**

題目形式：

- 手寫題可以用手寫拍照、打字的方式完成，但最後要**統一轉成.pdf檔**繳交  
檔名為HW4\_{學號}.pdf。例如：**HW4\_0123456.pdf**
- 程式題則繳交程式原始碼(.c檔/.cpp檔/.h檔 if need)  
檔名為HW4\_{題號}\_{學號}.c / .cpp。例如：**HW4\_4\_0123456.c / .cpp / .h**

繳交方式：

- 將上述**總共三個檔案及h檔(if need)**(手寫題pdf檔\*1+程式題c/cpp檔\*2)**直接上傳至e3**
- 檔名 / 格式錯誤者扣該次作業總分10分。
- 程式部分輸出格式請照作業說明，若不同會酌量扣分。

收作業規則：

- 遲交一個禮拜內分數打七折，超過一個禮拜即不接受補交。
- 遲交期限內僅接受原本沒交作業的同學補交，不接受先前交過作業的同學再次補交，若要修改答案請在繳交期限內修改完畢。
- 請務必重新整理，確認檔案已成功上傳至e3。

如有任何問題，麻煩從e3來信給所有助教。