109 Data Structure
the 3rd Homework

11月	1	2	3	4	5	6	7	9
	8	9	10	11	12	13	14	10
	15	16	17	18	19	20	21	11
	22	23	24	25	26	27	28	12
	29	30	1	2	3	4	5	13
	6	7	8	9	10	11	12	14
	13	14	15	16	17	18	19	15
12月	20	21	22	23	24	25	26	16
	27	28	29	30	31			17

黑圈：HW3的開始與結束
藍圈：HW4的開始與結束
橘圈：第二次小考
紅圈：上機考
1/5：期末考

A Sad Teaching Assistant (40%)

- There's a teaching assistant for a data structure course in a university.
- One day, the students told him that the homework he proposed is an algorithm-like homework. It's not a homework for a data structure course.

- He was depressed. Thus, he decides to propose a homework which is really associated to data structure.

- Finally, the TA I just talked about is not me.

A Sad Teaching Assistant (40%)

- 1) What the students need to do is a really simple task, and they will obtain 20pts. after doing this job.
- The TA gives the students (Note: the TA is not me.) N integers, I_1, I_2, …, I_n.

- Next, the TA gives the students several tuples of one command and two numbers, (c, x, y).
- If (c == 'M'), the students need to modify I_x to y.
- If (c == 'P'), the students need to print:
  - $\max(I_i + I_{i+1} + \ldots + I_j)$, $x \leq i \leq j \leq y$

A Sad Teaching Assistant (40%)

- Input:
  - Line 1: An integer T, indicates how many times the task will be asked for.
  - Line 2: An integer N, indicates the number of the integers TA will give.
  - Line 3: N integers, I_1, I_2, …, I_N, indicate the initial numbers.
  - Line 4: An integer M.
  - Next M lines: C x y tuples
- Output:
  - For each P x y tuple, you need to print
    - max(Ii + Ii+1 + … + Ij)

A Sad Teaching Assistant (40%)

- For example:
- 1 (T)
  - 5 (N)
    - 1 2 -100 3 4
  - 5 (M)
    - P 1 5 // print 7 (3+4 = 7)
    - P 1 2 // print 3 (1+2 = 3)
    - M 3 5 // modify the integers into 1 2 5 3 4
    - P 1 5 // print 15 (1+2+5+3+4 = 15)
    - P 2 4 // print 10 (2+5+3 = 10)

- output : 7\n3\n15\n10\n (\n : EOL)

A Sad Teaching Assistant (40%)

- For this problem:

- The times of the task might be asked for is T.
  - 1 <= T <= 5
- The number of integers given is N.
  - 1 <= N <= 50000
  - The given integers are [-10000, 10000]
- The number of command tuples given is M.
  - 1 <= M <= 50000

- Time limit: 2 second per data.

A Sad Teaching Assistant (40%)

- Hint: You should make sure every command could work within time complexity O(log n).

A Sad Teaching Assistant (40%)

- Of course, the task is so simple that every student makes it. The TA doesn't give up, and he quickly makes a second mission. (20pts.)

- There is no more 'M' command. However, the 'P' command becomes more complex:
- For a command C = x1 y1 x2 y2:
- You need to figure out
  - max(Ii + Ii+1 + … + Ij),
  - $x_1 \le i \le y_1,\ x_2 \le j \le y_2, while\ x_1 \le x_2, y_1 \le y_2$

A Sad Teaching Assistant (40%)

- Input:
  - line 1: An integer T, indicates how many times the task will be asked for.
  - line 2: An integer N, indicates the number of the integers TA will give.
  - line 3: N integers, I_1, I_2, …, I_N, indicate the initial numbers.
  - line 4: An integer M.
  - next M lines: x1 y1 x2 y2 tuples.
- Output:
  - For each P x y tuples, print
    - max(Ii + Ii+1 + … + Ij)

A Sad Teaching Assistant (40%)

- For example:
- 2 (T)
  - 6 (N)
    - 5 -9 -5 9 -3 4
  - 3 (M)
    - 1 3 6 6 // print 5 ((-5) + 9 + (-3) + 4 = 5)
    - 1 2 3 6 // print 1 (5 + (-9) + (-5) + 9 + (-3) + 4 = 1)
    - 4 4 6 6 // print 10 (9 + (-3) + 4 = 10)
  - 6 (N)
    - -2 -7 8 9 9 -5
  - 1 (M)
    - 6 6 6 6 // print -5 ((-5) = (-5))
- output : 5\n1\n10\n-5\n (\n : EOL)

A Sad Teaching Assistant (40%)

- For this problem:

- The task will be asked for T times.
  - 1 <= T <= 5
- The number of given integers is N.
  - 1 <= N <= 10000
  - The given integers are [-10000, 10000]
- The number of given command tuples is M.
  - 1 <= M <= 10000

- Time limit: 2 second per data.

Morse Code (60%)

- Morse code is a method used to encode text characters as sequences of dots and dashes.

字元	代碼	字元	代碼	字元	代碼	字元	代碼
A	.-	H	....	O	---	V	...-
B	-...	I	..	P	.--.	W	.--
C	-.-.	J	.---	Q	--.-	X	-..-
D	-..	K	-.-	R	.-.	Y	-.--
E	.	L	.-..	S	...	Z	--..
F	..-.	M	--	T	-		
G	--.	N	-.	U	..-		

Morse Code (60%)

- You receive confidential documents. Your supervisor used Morse code to encode the contents.

- However, there are no spaces separating the letters in the documents.
- Therefore, there may be several interpretations of any single decoded sequence.

- For example:
- “-....--.-.” could be: “DUC”, “DUTETE”, “BAC”, “BANN”, ...

Morse Code (60%)

- You try to decode the contents of the documents. Because there are too many documents, you decide to write a program to help you.
    - It’s hard for a machine to recognize which interpretation is reasonable. Thus, you use a dictionary to support this task.

Morse Code (60%)

- 1) English dictionary to Morse code. (15%)
- Input:
  - line 1: An integer N, indicates the number of words in the dictionary.
  - next N lines: A word.
  - line N+2: An integer M.
  - next M lines: a sequence of dots and dashes.
- Output:
  - For each sequence, you need to print:
    - It is found in the dictionary.
    - It is not found in the dictionary.

Morse Code (60%)

- For example:
- 4 (N)
  - APPLE // .-.--..--..-...
  - BANANA // -....--..--..-
  - CAT // -.-..--
  - DOG // -..-----.
- 2 (M)
  - -.-..-- //printf("-.-..-- is found in the dictionary.\n")
  - -.- //printf("-.- is not found in the dictionary.\n")

Morse Code (60%)

- 2) Determine if two words have the same prefix. (15%)
- Input:
  - line 1: A word w1.
  - line 2: An integer N.
  - next N lines: A word w2.
- Output:
  - For each w2, you need to determine:
    - After w1, w2 are both encoded into Morse code:
      - w2 is the same as w1.
      - w2 has the same prefix as w1.
      - w2 doesn’t have the same prefix as w1.

Morse Code (60%)

- For example:
- CAT (w1) // -.-..--
- 3 (M)
  - NDTT // -.-..--
  - KIT // -.-..-
  - RUT // .-...--

- output:
  - “NDTT is the same as CAT.”
  - “KIT has the same prefix as CAT.”
  - “RUT doesn’t have the same prefix as CAT.”

Morse Code (60%)

- 3) With the dictionary and the program, you only need to check the part of the interpretations. (30%)

- Input:
  - line 1: A Morse sequence with a maximum length L.
  - line 2: An integer N indicates the number of words in the dictionary.
  - next N lines: One word.
- Output:
  - The number of messages R, which are possible to generate with the Morse sequence and the dictionary.

Morse Code (60%)

- For example:
- ......-...-..---.-----.-..-..-..
- 5
- HELL // ......-...-..
- HELLO // ......-...-..---
- OWORLD // ---.-----.-..-..-..
- WORLD // .-----.-..-..-..
- TEST // -....-

- output: 2 (HELL+OWORLD, HELLO+WORLD)

Morse Code (60%)

- The Morse sequence has a maximum length L.
- 0 < L < 100000
- The number of words in the dictionary is N.
- 0 < N < 100000
- The words in the dictionary have a maximum length M.
- 0 < M < 20

- Time limit: 2 second per data.

Reminders

- For all of the question, please read test.txt as input and write output.txt as output.
- 對於所有問題，請都讀test.txt作為input、寫output.txt作為output
- If you can, please let me know how to change your I/O file name so that I can modify the path from test.txt to test1.txt, test2.txt, etc.
- 假如可以的話，讓我清楚知道從哪裡更改你I/O檔案的名稱，方便我可以從讀test.txt改成讀test1.txt, test2.txt，會改得比較快。
- 如果我看不懂，那我不會改你的code，一律讀test.txt。

Reminders

- Only accept C
- Deadline : 2020/12/07 23:59, please be on time.
- File name : [student ID]_[question No.(1or2)]-[sub question No.(1,2,3)].[file name extension]
- e.g. 7109056193_1-1.c
- If there are more than 1 file for 1 question, please give a readme.txt for me and let me know the meaning of each file.
- No need to give me the output, I'll execute your program.

- Zip all your files and hand in on the i-learning, the file name is [student ID]_homework3.
- Plagiarism is prohibited.
- Dev-C++ 5.11 is used for checking this homework.