Programming Design, Spring 2016
Lab Exam 3

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

For all the problems in this exam, you are allowed to use any technique.

Problem 1

(30 points) Given n integers $(x_1, x_2, ..., x_n)$, determine whether there exist a value $k \in \{1, 2, ..., n\}$ such that $x_1 + \cdots + x_k = x_{k+1} + \cdots + x_n$. If yes, output the largest k and the sum of all the n numbers; otherwise, output 0 and the sum of absolute values of all the n numbers.

Input/output formats

There are 15 input files. In each file, there are two lines of integers. The first line contains one integer n. The second line contains n integers x_1, x_2, ..., and x_n. Two consecutive numbers in a line are separated by a white space. It is know that $n \leq 1000$ and $-100 \leq x_i \leq 100$. To output the result, separate the two integers by a white space. For example, the following input

```
11
1 5 -4 4 6 8 -2 3 9 8 -2
```

requires you to print out

```
7 36
```

Grading criteria

30 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.

Problem 2

(30 points) You have a board with n rows and n columns, where the square at the intersection of row i and column j is called square (i, j). m coins are put in squares (i_1, j_1), (i_2, j_2), ..., and (i_m, j_m), where no two coins are put in the same square. Two squares are adjacent if they share one edge. For example, squares $(4, 5)$ and $(4, 6)$ are adjacent but $(4, 5)$ and $(5, 6)$ are not. A coin is isolated if no coin is adjacent to it. Find the number of isolated coins. Then find the number of non-isolated coins that are not at the border of the board (i.e., having four adjacent squares).

Input/output formats

There are 15 input files. In each file, there are $m + 1$ lines. The first line contains two integers n and m. It is known that $1 \leq m \leq n^2 \leq 100$. The kth line contains two integers i_{k-1} and j_{k-1}, $k = 2, ..., n + 1$. It is known that $1 \leq i_k \leq n$ and $1 \leq j_k \leq n$, $k = 1, ..., n$. Two consecutive numbers are separated by a white space. To output the result, separate the two integers by a white space. For example, the following input

```
4 7
1 1
1 3
3 2
2 3
2 4
3 4
4 1
```

requires you to output

```
3 1
```

because there are three isolated coins (in squares $(1,1)$, $(3,2)$ and $(4,1)$) and one non-isolate coins not at the border (in square $(2,3)$).

Grading criteria

30 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.

Problem 3

(40 points) In a crossword puzzle, English words are put vertically or horizontally on a board consisting of n rows and m columns. Each character of a word occupies a square. If two words share the same square (if this happens, it must be that one word is placed vertically while the other is placed horizontally), that square must contain the same character. Below is an example.

a	p	p	l	e		
			i			
	t	w	o			
			n	o	w	
					a	m
					y	

You will be given an n by m board, k words, the starting position of each word, and whether each word is placed vertically or horizontally. The starting position is given as a pair of integers (x, y), which means the square at the intersection of row x and column y. For example, in the above example the word "way" has its starting position at $(4, 6)$. Note that a word may be placed only in a top-to-down or left-to-right direction. Therefore, as long as you know "way" is placed at $(4, 6)$ vertically, it must occupy square $(4, 6)$, $(5, 6)$, and $(6, 6)$. It cannot occupy squares $(4, 6)$, $(3, 6)$, and $(2, 6)$.

You need to check whether a given input form a correct answer, i.e., whether there is no square containing two different characters. If there is no conflict, output the number of total occupied squares; otherwise, output the number of total non-occupied squares plus the number of conflicting squares (the squares that contain two different characters).

Input/output formats

There are 20 input files. In each file, there are $k + 1$ lines. The first line contains three integers n, m, and k. It is known that $1 \leq n \leq 100$ and $1 \leq m \leq 100$. Each of the second to the $(k + 1)$th lines containing an English word (with no white space or special symbols), an integer x, an integer y, and an integer d. This means that the word is placed at square (x, y) in direction d, where $d = 1$ means vertical and $d = 2$ means horizontal. It is guaranteed that this word can be put on the board without breaking the boundary. It is known that (x, y) is a valid position and $d \in \{1, 2\}$. Two consecutive values are separated by a white space. To output the result, print out an integer directly.

For example, the following input

```
6 7 6
apple 1 1 2
two 3 2 2
now 4 4 2
am 5 6 2
lion 1 4 1
way 4 6 1
```

gives us exactly the example above. In this case, the output should be

```
15
```

which is the number of total occupied squares.

If the last input line becomes

```
why 4 6 1
```

this set of words form an infeasible solution. In this case, the output should be

```
28
```

which is the number of non-occupied squares (27) plus the number of conflicting squares (1).

Grading criteria

40 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.