
CS29003 ALGORITHMS LABORATORY
ASSIGNMENT 3
Date: 2nd Sep, 2021

Important Instructions

1. **Input:** To be taken from command line
 2. **Format fo files to be submitted to Moodle and HackerRank:** ROLLNO_A3.c/.cpp
 3. Take inputs from command line in the specified format
 4. You are to **stick to the file input output formats strictly** as per the instructions.
 5. Submission through **.zip files are not allowed.**
 6. Write your name and roll number at the beginning of your program.
 7. Do not use any global variable unless you are explicitly instructed so.
 8. Use proper indentation in your code.
 9. Please follow all the guidelines. Failing to do so will cause you to lose marks.
 10. **There will be part marking.**
-

Divide and Conquer

Problem Statement

You are an army commander of ALGOLAND. Your country is currently at war with BRUTEFORCE - a neighbouring country. The army of BRUTEFORCE is approaching towards SADDLE POINT- a strategic location in this war. The main assignment today for you is to take over SADDLE POINT in order to gain a strategic advantage over your enemy.



Figure 1: A helicopter carrying solders to SADDLE POINT

To takeover this location you have to place your soldiers around this location. The SADDLE POINT is a square area having $N \times N$ cells where $N = 2^k$ ($N \leq 2^{10}$). Each cell is a $5m \times 5m$ area which can be

covered by a single soldier. You have also got an information from the intelligence team that one of these cells is a mine field. You know this cell location and cannot place any of your soldiers in this cell and have to keep it empty. The army is divided into M battalions where $0 < M \leq 26$. Each battalion contains a certain number of soldiers and is denoted by a character β where β is one of the letters in alphabet (a-z). The rank of the battalion is lexicographic position of the character associated with the battalion (i.e. the a-battalion will have a higher rank than the b-battalion).

A battalion is again divided into units where a unit comprises of 3 soldiers. In order to place the soldiers in the SADDLE POINT, a helicopter is assigned which can carry only one unit from a single battalion at a time. Based on your directions, it drops the soldiers in a 2×2 square area, and the soldiers in the unit cover this small square area, leaving one of the cell unoccupied as shown in Figure 2. While choosing a unit you have to make sure that the battalion selected has the most number of soldiers left. If two battalions have the same maximum number of soldiers, you have to select the battalion which has a higher rank. In order to gain the strategic position you have to fill all the $N \times N$ cells with your soldiers except the mine field. Can you propose an algorithm to fill the SADDLE POINT using a divide and conquer approach ? Best of luck on your assignment, Commander!!

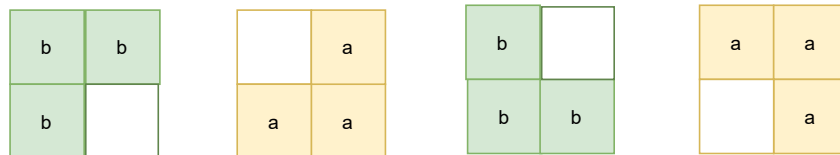


Figure 2: Units from different battalions covering a 2×2 area. Each soldier is represented by his/her battalion symbol.

Input

The first line of the input contains N which is the dimension of the square representing SADDLE POINT. The second line contains the x, y location denoting the cell which is a minefield. The third line contains M which is the total types of battalion available that day. Each of the next M lines contains a unique alphabet β out of a to z denoting a battalion and an integer num which denotes the number of units in this battalion. It is given that you will always have sufficient soldiers to fill the whole area.

$$N = 2^k \quad \text{where} \quad 0 < k < 10 \quad (1)$$

$$0 < M \leq 26 \quad (2)$$

$$0 \leq x, y < N \quad (3)$$

$$0 < \text{num} \leq \text{INTMAX} \quad (4)$$

Output

There two variants of the output for HackerRank and Moodle. In **HackerRank**, you have to print M lines where each line mentions how many units are left in β -battalion by printing a statement "There are x units left in β -battalion". The order of the M lines should follow the order given in the input.

Note: While the output of HackerRank may be achieved by a trivial algorithm, you have to make sure your code is following a divide and conquer algorithm to get the marks for passing the test cases.

For **Moodle**, the output should print two matrices of $N \times N$ dimension one after the other. The first matrix is the strategy which shows the final position of the soldiers after placing them. The other matrix is sequence which shows the sequence and position in which each unit was placed. Finally, you have to print M lines where each line mentions how many units are left in β -battalion by printing a statement “There are x units left in β -battalion”. The order of the M lines should follow the order given in the input.

Example

In Figure 3 (a), we show the initial configuration of the SADDLE POINT area and the 2 battalions to fill the matrices. In the first step we start from the top left corner, find that it is a invalid and move to the next cell in the same row. The next cell is empty and we place a unit as shown in figure 3 (b). Which one will be the next unit to be placed ? What will be an appropriate position to place the unit in the first step?

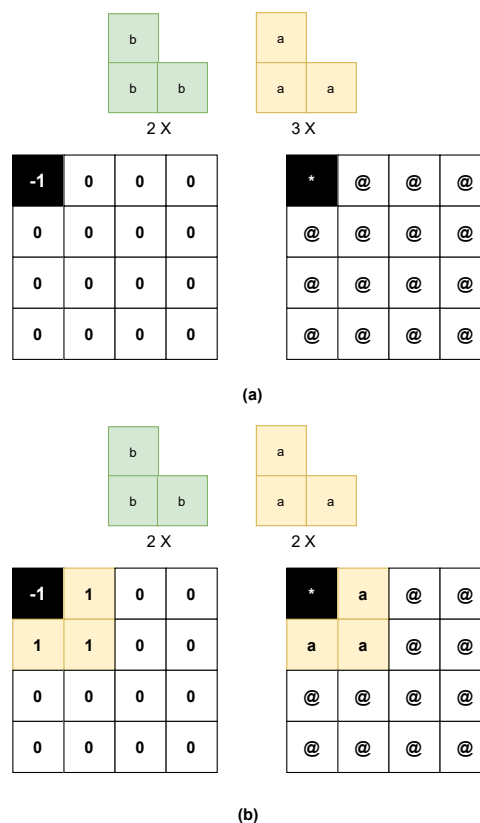


Figure 3: (a) The initial configuration of both the matrices represent SADDLE POINT, there are 3 units in a-battalion and 2 in b-battalion (b) We choose a unit from the a-battalion because it had the most number of units and place it in the position shown. $-1/*$ show the position of the cell with mines.

The next sections shows the input, the output for moodle and the output for HackerRank for some of the test cases.:-

Test Case 1

Input

FILE: *Sample/input1.txt* _____

2
1 1
2
a 3
b 9

Output Moodle

FILE: *Sample/output10.txt* _____

b b
b *

1	1
1	-1

There are 3 units left in a-battalion
There are 8 units left in b-battalion

Output HackerRank

FILE: *Sample/output11.txt* _____

There are 3 units left in a-battalion
There are 8 units left in b-battalion

Test Case 2

Input

FILE: *Sample/input2.txt*

```
4
2 3
3
a 3
b 6
c 9
```

Output Moodle

FILE: *Sample/output20.txt*

```
c  c  c  c
c  c  c  c
b  c  c  *
b  b  c  c
```

3	3	2	2
3	1	1	2
4	1	5	-1
4	4	5	5

```
There are 3 units left in a-battalion
There are 5 units left in b-battalion
There are 5 units left in c-battalion
```

Output HackerRank

FILE: *Sample/output21.txt*

Test Case 3

Input

FILE: *Sample/input3.txt*

```
8
5 4
4
a 100
b 100
c 100
d 100
```

Output Moodle

FILE: *Sample/output30.txt*

```
a a d d d d c c
a c c d d b b c
b c c c a a b b
b b c a a a b b
b b a a c c b b
b d a a * c a b
c d d d d a a a
c c d d d d a a
```

9	9	8	8	4	4	3	3
9	7	7	8	4	2	2	3
10	7	11	11	5	5	2	6
10	10	11	1	1	5	6	6
14	14	13	1	19	19	18	18
14	12	13	13	-1	19	17	18
15	12	12	16	20	17	17	21
15	15	16	16	20	20	21	21

```
There are 94 units left in a-battalion
There are 95 units left in b-battalion
There are 95 units left in c-battalion
There are 95 units left in d-battalion
```

Output HackerRank

FILE: *Sample/output31.txt*

```
There are 94 units left in a-battalion
There are 95 units left in b-battalion
There are 95 units left in c-battalion
There are 95 units left in d-battalion
```

To test your own code, take the input from command line similar to formatting specified in the test cases. HackerRank will have some test cases so you can directly submit the code in Hackerrank to run on those test cases. HackerRank test cases are only provided to check the correctness of the code partially. You have to create additional test cases to handle corner cases (if any) and check the efficiency of the code.

Implementation details

We will be providing some of the functions to help you with your assignment. `setup_2d_array()` will setup both the matrices. The functions `print_array()` and `print_soldiers_left()` will print the output in the appropriate format. The structure `different_battalion` is also provided. You are free to define structures, classes and functions of your own (thus, you can choose not to use the given functions) but make sure to maintain the input/output formats.

Your sample submission should be as follows:

```
//20CS30004_A3.c/cpp
#include <stdio.h>
#include <stdlib.h>
//for c++
#include <iostream>

struct battalion{
    char beta;
    int num;
};

/*code for setup_2d_array, print_array and print_soldiers functions will be given in the HackerRank
but you may or may not use them*/

int main()
{
    // add your codes to take inputs
    /* use the structure type_battalion and create a array "different_battalions" to
       get the types and number of units per type */
    setup_2d_array(strategy,sequence,x,y,n);
```

```
// write your code to place the soldiers

// print the output of this program
/*uncomment the following two lines when submitting to
moodle/testing your code on custom input*/
//print_array(strategy,sequence,n);
//printf("\n");
//you only need to print this when submitting the code to HackerRank.
print_soldiers_left(different_battalions,m);
return 0;

}
```
