Indian Statistical Institute

Semester-I 2019-2020

M.Tech.(CS) - First Year

Lab Test I (16 August, 2019)

Subject: Data and File Structures Laboratory

Total: 60 marks Duration: 4 hrs.

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `cs19XX-test1-progY.c`
   IMPORTANT: Insert a single alpha-numeric string of your choice, 6-8 characters long, in the name given above as shown in the examples below. Think of this string as something like a security password, except that you are not required to remember the string. Examples: `cs1940-assign3-x19jdh4-pr og1.c`, `ppo03wws-cs1940-assign3-prog2.c`, `cs1940-assign3-prog2-jsiwm7de .c`
2. When you have finished, copy all your files to `~dfslab/2019/labtest1/cs19xx/` on 192.168.64.35.

1. **(20 marks)** Suppose you are given some text, consisting of *words*, separated by one or more non-word characters. For this problem, define a *word* to be any maximal, non-empty sequence of Roman letters (`A`–`Z`, `a`–`z`). Write a program to find the word-length (the number of characters in a word) that occurs most frequently in the input, along with its frequency. For example, if your input consists of

   - 10 words that are 2 characters long
   - 17 words that are 3 characters long
   - 24 words that are 4 characters long
   - 2 words that are 10 characters long

   and no other words of any other length, your program should print `4 24` as the output.

   Input format: Your program should read the text from standard input.

   Output format: The word-length that occurs most frequently in the given input, along with its frequency.

   Sample input 1

   ```
   a xyz, !@*% ``foo bar'' !jkl:xyz uvw-pq-mnzc
   742 b742 8i8
   ```

   Sample output 1

   ```
   3 6
   ```

   Explanation: The words in the input are `a xyz foo bar jkl xyz uvw pq mnzc b i`. Thus, the input has 3 words of length 1, 1 word of length 2, 6 words of length 3, and 1 word of length 4.

Sample input 2

```
Madam I'm Adam
```

Sample output 2

```
1 2
```

Sample input 3

```
;+-)~^ :{& *%$
```

Sample output 3

```
0 0
```

NOTE: If multiple word-lengths have the same maximum frequency of occurrence, you may print any one length, along with its frequency. For example, if the input contains 102 words of length 2, 102 words of length 5, and less than 100 words of other lengths, your program should print either `2 102` or `5 102`.

2. **(20 marks)** Recall how a group of children playing hide-and-seek choose IT (the *chor*). They stand in a circle, and one child (called the *counter*, say) recites some kind of a rhyme consisting of m words, starting with himself / herself and moving on to the next child after reciting each word in a circular fashion. The child at whom the rhyme ends is eliminated from subsequent rounds of counting. Counting for the next round starts from the child following the eliminated child (in circular order). The process continues until one child remains, who becomes IT.

   Write a program to simulate the above activity, and to determine the order in which children are eliminated.

   Input format: An integer (say n) specifying the number of test cases, followed by n lines, each of which corresponds to a test case. These lines will each consist of 3 integers, corresponding to the total number of children, the serial number of the child chosen as the counter, and the length of the rhyme, respectively. Note that numbering of children starts from 1.

   Output format: Your program should print the serial numbers of the children in the order in which they are eliminated, and finish with the number of the child who becomes the *chor*. For each test case, your program should print the output in a single line.

   Sample input 1

   ```
   3
   4 1 10
   5 2 6
   5 2 3
   ```

   Sample output 1

   ```
   2 3 1 4
   2 4 3 1 5
   4 2 1 3 5
   ```

3. **(20 marks)** Consider a rectangular area that is $N \times M$ in size (N and M are positive integers). You have an unlimited supply of square tiles of size $2^i \times 2^i$, where $i = 0, 1, 2 \ldots$. Write a program to compute the **minimum** number of tiles required to cover the given area. Note that N and M need not be of the form 2^k for some integer k.

   Input format: N and M will be provided in that order via stdin.

   Output format: The minimum number of tiles required to cover the area (as computed by your program) should be printed to stdout.

   Sample input 1

   ```
   1024 256
   ```

   Sample output 1

   ```
   4
   ```

   Sample input 2

   ```
   1025 256
   ```

   Sample output 2

   ```
   260
   ```

   Sample input 3

   ```
   3 100
   ```

   Sample output 3

   ```
   150
   ```