Coursework Task

Higher Computing Coursework Task 2011–2012

Introduction

Suretest employ people to test computer games. New employees are given a budget to buy a suitable computer system for testing the games. Each tester plays the games on their own desktop computer system and logs the faults they encounter while playing each game. To help collate the test data generated by their testers, Suretest wish to create a program to store and process test results.

Part 1

Suretest decide to commission a software development company to develop a program to log faults found by their testers. The program will be tested on the game, "Hamster Commanders".

The software will

- Input the game levels which crashed during each test session.
- Count the number of times each level has crashed.
- Produce a list of the number of crashes in each level.
- Calculate which level/levels had the greatest number of crashes.

The data below should be entered into the programme.

Test Run	Levels which Crashed
1	e,e,d,b,d,b,a,a,a,e
2	e,f,a,a,a,b,d,a,b,e
3	d,e,d,c,d,a,f,f,e,a

How the program should work

Entering the log data

Hamster Commanders has 6 levels (a to f). The data should be entered as a 10 character string.

For example the data for test run 1 should be entered as "eedbdbaaae".

Displaying the total crashes for each level

The total crashes should be displayed as shown below.

```
Level               Number of Crashes
a                   3
b                   2
c                   0
d                   2
e                   3
f                   0
```

Displaying the level/levels which crashed the greatest number of times

The information on the level/levels with the most crashes should be displayed as shown below.

```
The levels with the most number of crashes were:
Level a
Level e
These levels crashed 3 times
```

Main Algorithm

1. Get valid test run data
2. Count and display the total number of crashes for each level
3. Find greatest number of crashes and display the levels where they occurred

What you have to do:

	Tasks	Evidence required	Marks
1	Indicate data flow on the main algorithm.	Algorithm with data flow.	2
2	Refine step 1.	Pseudocode for step	1
3	Refine step 2.	Pseudocode for step	3
4	Refine step 3.	Pseudocode for step	3
5	Using a software development environment of your choice, implement the algorithm. Use separate sub-programs where appropriate. Use parameter passing where appropriate.	Listing of implemented program.	16
6	Test the program with the data provided.	Hard copy of results for three given test runs.	1
7	Test the program with your own test data to ensure robustness.	Hard copy of your test results and report on test results.	2
8	Evaluate maintainability.	Brief report on maintainability of program code.	2