

1806ICT Programming Fundamentals – 2023

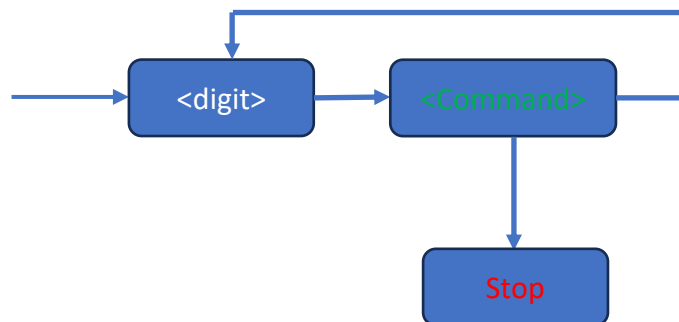
In-Trimester Test – 2 hours

If you have any trouble understanding a question then you are allowed to ask for clarification via the meeting Teams Chat or verbally using your microphone.

Any C code found to have been copied from the Internet will result in the appropriate actions being taken.

1. (25 Marks) Write a C program that contains a function `sum()` that takes a `C string` as a parameter. The format of the `C string` is shown in the following diagram and the original input is from the command line. The function `sum()` uses a `bool` return value to indicate whether the command was valid and also returns the result as a parameter.

The valid syntax of the input is:



Where `<Command>` is given by:

`<* or / or % > <digit>`

The `sum()` function evaluates each `<digit><Command>` pair and adds the result to the total. So a valid sequence for running the program is: `./a.out 1*8 9/4` and would give the result 10 whereas `./a.out 1*8 2/0` would give "Invalid Input".

You must start with the supplied file Q1.c and only add the necessary changes.

The output from a correct version of the program is:

```
Result of 1*8 5/4 is 9
6*8 2/0 Invalid Input
Result of 9%8 6*4 2/4 is 25
1&3 Invalid Input
```

2. (25 Marks) Write a C program that accepts the same input as the Assignment – Milestone 1 and generates integers using the function supplied for the assignment (do not store the counts). Implement a technique that reduces the size of the array by changing how duplicates are stored.

You must start with the supplied file Q2.c and only add the necessary changes.

Sample output from a version of the program is shown below. Note that because of the use of different random number generators your numbers may not match exactly:

```
./a.out 1000000 1 100
```

Reduced from 1000000 to 200, a reduction of 99.98%

`./a.out 1000000 1 1000`

Reduced from 1000000 to 2000, a reduction of 99.80%

`./a.out 1000000 1 10000`

Reduced from 1000000 to 20000, a reduction of 98.00%

`./a.out 1000000 1 100000`

Reduced from 1000000 to 199948, a reduction of 80.01%

`./a.out 1000000 1 1000000`

Reduced from 1000000 to 896763, a reduction of 10.32%

`./a.out 1000000 1 10000000`

Reduced from 1000000 to 998501, a reduction of 0.15%

3. **(25 Marks)** Write a C program that stores words read from a dictionary file so that they can be accessed quickly (not sequential search). Given a word on the command line, the program lists all words that are close to that word where two words are close if they are different, contain the same letters but the order of two adjacent letters is reversed in one and both words are in the dictionary.

You must start with the supplied file Q3.c and only add the necessary changes.

Sample output from a correct version of the program is:

`./a.out dictionary.txt united`

untied is close to united

`./a.out dictionary.txt nuclear`

unclear is close to nuclear

`./a.out dictionary.txt trials`

trails is close to trials

`./a.out dictionary.txt three`

there is close to three

`./a.out dictionary.txt accreditation`

There are no words close to accreditation

`./a.out dictionary.txt xyz`

xyz is not in the dictionary

On completion the three files, Q1.c, Q2.c, Q3.c must be zipped into a zip file titled <number>_<Last Name>.zip which is the file to be submitted.

Submission to L@G or via email to w.pullan@griffith.edu.au must be before 2:10pm unless notified otherwise during the test.