

CS1020E: DATA STRUCTURES AND ALGORITHMS I

Practical Exam 2 - 😊

(Week 12, Friday, 4 November 2016)

Instructions

1. The TA will login to windows for you before the session
If you reboot your PC, ask the TA to login for you again
2. You will be given a new UNIX account for the practical exam
Only use this account to develop your program



Open book (hardcopy)



1hr 45min + 10min

3. The **ONLY programs you are allowed to open** are SSH Secure Shell Client, and the local C++ references
You may open more than one window of each. Opening any other program is cheating
4. Do **NOT develop your program in additional files**
Do **NOT change the filename / location of existing files**
5. Ensure you **enter your full name, user ID** (starting with 'e') and **login id of the new account** given to you in the comments at the **top of each of file** you code in
6. At the end of the session, logoff but do not shutdown
You are NOT free to leave until the TA tells you to do so
Remain quietly in your seats. This is an official assessment
7. If you have any **question, raise your hand** and the TA will attend to you
Attempting to communicate with others is cheating; disciplinary action will be taken against you
8. If you lock your screen using Ctrl+S, press Ctrl+Q to unlock it
If you move your job into the background (Ctrl+Z), type fg to bring it back into the foreground
i.e. Do NOT press Ctrl+Z
If your program runs in an infinite loop or takes too long to complete, press Ctrl+C twice to terminate
To simulate end-of-file when entering input from the keyboard, press Ctrl+D

Advise

- Save, compile, and test your program frequently
- Few functionalities work > Program with all functionalities half done > Non-compile-able code
- With good design and coding style, you will help yourself
- Design system and algorithm **completely** on paper, prove that it works, before coding

Testing

All the input files are provided in your plab account. However, we only give you the output of `fish1.out` and `monkey1.out`, which are exactly the same as the sample input/output shown in this paper. You are advised to **create your own test data files**.

Compiling your program:

```
g++ -std=c++11 fish.cpp
g++ -std=c++11 monkey.cpp
```

Running with stdin and stdout redirected:

```
a.out < fish1.in > 1.a
a.out < monkey1.in > 1.a
```

Comparing output:

```
diff fish1.out 1.a
diff monkey1.out 1.a
```

Submission Instructions

Your home folder contains two subdirectories:

- `ex1/` - Write your **program ONLY in the file `fish.cpp`**
- `ex2/` - Write your **program ONLY in the file `monkey.cpp`**

Other files will NOT be uploaded to CodeCrunch. Use the `cd` command to change directory.

Therefore, code your programs in the **skeleton files provided**. **If you do not see a skeleton**, it means you are writing your program in the **wrong place**, it will not be transferred, and you will receive heavy penalty.

Grading

Ex 1

15% for coding style and design, **conditional** on attaining at least 10% for correctness:

- Naming of identifiers, appropriate data types
- Proper variable scope and modularity
- Meaningful and appropriate comments
- Indentation

35% for correctness, with partial marking:

- Base cases
- Recursively calling smaller problem(s)
- Handling of result of smaller problem(s)
- Viewing recursive part of problem as simply as possible

Ex 1 implementation notes:

- You **MUST** use **recursion** to solve the bulk of the problem
- Your recursive function should get its input and output as parameters and/or return value
- Do **NOT** rely on **global variables** or attributes
- When calling your function, `main()` should **only pass** whatever it needs to be **concerned** with

Ex 2

15% for coding style and design, **conditional** on attaining at least 10% for correctness:

- Naming of identifiers, appropriate data types
- Meaningful and appropriate comments
- Indentation

35% for correctness:

- 5% for each of the 7 test cases, **binary** grading
- Your solution has to be **efficient**, or larger test cases will fail due to "Time Limit Exceeded"

Ex 2 implementation notes:

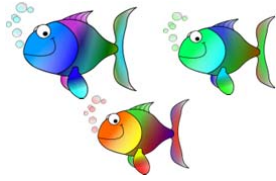
- You may use any C++ code that works to solve Exercise 2
- You may write your entire program in `main()`, but observe good coding habits aside from that

Warnings:

- ex1 is on **recursion**, while ex2 is on developing an **efficient algorithm** - don't mix them up
- **Non-compile-able code** will result in (total marks \neq 2); no excuses will be entertained
- **Commented-out code** will be ignored
- There will be **heavy penalty** if your program does not end up being transferred onto CodeCrunch, or we cannot identify your program due to missing information at the start of each file

Exercise 1 - Fish

Mer



Sample Input

☺

Sample Output

☺

Explanation

Exercise 2 – *Monkey*

In M



Sample Input

☺
☺

Sample Output

☺
☺
☺
☺
☺
☺

Explanation

Efficiency

- The first test case is this sample input/output
- The next 4 test cases have $n \leq 1\,000$, worth 20% of PE2 marks
- The last 2 test cases have $n \leq 10\,000$, worth 10% of PE2 marks