CS2040 Lab 2

AY19/20 Sem 2, Week 4

One-Day Assignment 0 – Pea Soup

Algorithm released on LumiNUS under Labs > Solutions

- Future one-day assignments will have algorithms released on LumiNUS at 6pm on the Friday it is released, under Labs > Week X
  - Contents of the “Week X” folder will be moved to the main Labs/Solutions folders after the assignment is due, to avoid too much clutter

One-Day Assignment 0 – Pea Soup

Should read in input correctly

- Due to the way sc.nextInt() and sc.nextLine() works, running an sc.nextLine() immediately after an sc.nextInt() may result in it reading in an empty string
- Use an additional dummy sc.nextLine() to clear away the empty string first i.e.

```
int n = sc.nextInt();

sc.nextLine();

String input = sc.nextLine();
```

One-Day Assignment 0 – Pea Soup

Each time we process a new restaurant, have two flags (boolean variables) to indicate whether this restaurant has pea soup, or pancakes

Set the flag to true when pea soup or pancakes is encountered

If a restaurant has both set to true, output this restaurant and exit

Lab 2 – Buffered I/O

Scanner has convenient functions nextInt(), nextDouble() etc.

- Is actually pretty slow

Similarly, System.out.print()/println()/printf() may use up a lot of time if called repeatedly

Other faster methods of handling I/O functionality exist but are a bit more complicated to use

Lab 2 – Buffered I/O

Some take home assignments will require the use of buffered I/O (using Scanner/System.out will result in a Time Limit Exceeded verdict)

- All one-day assignments are guaranteed to be solvable just by using Scanner/System.out
- Optimization may be necessary in other parts of your program

Lab 2 – BufferedReader

Provides a much faster way to read in input

Initialise using the following line (be sure to import java.io.* first)

- `BufferedReader br = new BufferedReader(new InputStreamReader(System.in));`

Provides very few methods to read in input; the most frequently used one would be readLine(), which behaves much like Scanner's nextLine()

Lab 2 – BufferedReader

Method name	Description	Time
readLine()	Reads until it reaches the end of the line	O(N)

Lab 2 – PrintWriter

Provides a much faster way to write output

Basically the same as System.out methods, but delays printing until a .flush() or .close() is called (to avoid repeated switching between printing and computation, thereby saving some time)

Initialise using the following line (be sure to import java.io.* first)

- `PrintWriter pw = new PrintWriter(new BufferedWriter(new OutputStreamWriter(System.out)));`

Always call .flush() or .close() on the PrintWriter before exiting your program, or some output may not be printed

Lab 2 – PrintWriter

Method name	**Description**	**Time**
.print(String str)	Prints *str*	O(N)
.println(String str)	Prints *str*, followed by a newline character (‘\n’)	O(N)
.printf(String str)	Emulates the printf function of C	O(N)
.flush()	Flushes the buffer (ie. actually prints the contents of the writer to the screen)	O(1)
.close()	Calls flush(), then closes the writer. The writer cannot be used again	O(1)

Lab 2 – Kattio.java

Kattis provides its own version of a buffered I/O, which uses the classes from earlier

For input, it provides its own methods, covered in the next slide

For output, it uses the same methods as PrintWriter

Found at https://nus.kattis.com/help/java - search for “Kattio.java”

- Direct link to the file not provided, as the link may change if the file is updated

Not part of the standard Java API, but you can use it by copy-pasting it into your own program, or submitting the file alongside your own program to Kattis

Lab 2 – Kattio.java

Method name	Description	Time
.getInt()	Reads the next token in the input as an integer	O(N)
.getLong()	Reads the next token in the input as a long	O(N)
.getDouble()	Reads the next token in the input as a double	O(N)
.getWord()	Reads the next token in the input as a String	O(N)

One-Day Assignment 1 – T9 Spelling

Reminder: One-Day Assignments (from assignment 1 onwards) should be submitted on nus.kattis.com, not open.kattis.com and will be graded

Submission deadline for all One-Day Assignments is 11:59pm of the same day.

Simulating old text systems for phones

Each character has an associated series of button presses (eg. ‘a’ requires one press of the ‘2’ button, ‘b’ requires 2 presses and ‘c’ requires 3, while ‘d’ is a single press of the ‘3’ button) and so on

Need to pause if typing two consecutive characters that both use the same button (eg. ‘h’ and ‘i’ both use the ‘4’ button, so a pause is needed between the two)

One-Day Assignment 1 – T9 Spelling

Hint: every character is actually represented as an integer from 0 to 255 (ASCII value)

‘a’ has an integer value of 97, ‘b’ has an integer value of 98 etc.

Possible to “simulate” a dictionary (for Python/JavaScript users) of characters by creating an array of size 256, and using the character as the index

```
String[] arr = new String[256];

String input = sc.nextLine(); // assume input is “cd”

arr[‘c’] = “222”;

char letter = input.charAt(0);

System.out.println(arr[letter]); // prints 222
```

FastIO

```
FastIO fio = new FastIO();
int a = fio.nextInt(); // Read an int
long b = fio.nextLong(); // Read a long
double c = fio.nextDouble(); // Read a double
String d = fio.next(); // Read the next token as a String
String e = fio.nextLine(); // Read the next line
// Do not need to call dummy nextLine() after nextInt(),
nextLong(), nextDouble()
fio.print("Hi"); // Print Hi
fio.println("Hello"); // Print Hello with a trailing newline
fio.close(); // Flushes the rest of the output and closes the
FastIO instance
```

Telegram Group

https://t.me/joinchat/Fdw-UhDTnHmMggC4J2qpTg