Strings

This week is all about strings and string manipulations. A string is a sequence of characters. In C++, you implement them by declaring an array of characters while Java has an inbuilt String class. Like an array, strings are *zero indexed*, where in the character positions start from 0. In order to retrieve a character at a certain index, one only has to do type `string[index]` in most languages.
For example, if `string = "apple"`, then to get the character at the third position (index=2), `string[2] = 'p'`.

String manipulations mostly fall into two categories - finding subsequences of a string or comparing two strings.

Finding subsequences

Subsequences in a string are smaller sequences in a bigger string. These types of questions typically ask you to cut or splice a string from a bigger string to a smaller one. For example, substring is a special subsequence method and `substring(1,3)` of `'abcd'` gives `'bcd'`. Most languages have inbuilt functions that return substrings on providing a start and end index.

The questions may be made complex by asking you to find the number of possible substrings or get distinct substrings. You could also be asked to make a certain string following rules of some (given) pattern [read: steps of an algorithm] and then to output a certain substring of the produced string.

However, at the base, substring problems remain the same. You need to output a certain portion of a bigger string.

Comparing strings

As the name suggests, these types of questions usually ask you to compare two strings. Strings are a sequence of characters and the characters are stored in memory as numbers, with a unique number assigned to each character. A computer only understands only numbers and cannot differentiate between `a` and `@`. Hence, the ASCII code was developed. Though originally only 128 characters were assigned, modern computers use unicode which include characters from around the world.

Comparison in strings hence becomes simple arithmetic. In most languages, you can do basic math ops (`+`, `-`, `*`,`/`,`%`) simply by using the characters instead of numbers. For example, `cout << ('b'-'B');` will output `32`. This is because the ASCII code of `'B'` is `66` while the ASCII code of `'b'` is `98`.

Most languages also include inbuilt functions to compare to strings and to compare two strings while ignoring case. You can read more about them and other string functions on the official documentations of C++, Java, and Python3.

Parsing

Another kind of string manipulation is known as parsing. More than manipulation, it is an understanding of the input string into some framework or structure. It is something that humans do constantly. In fact, you are parsing while reading this handout.

Parsing can be done in a variety of ways. For example, a directory path such as `/usr/local/bin` can be parsed into a linked list as `'usr'` -> `'local'` -> `'bin'`. One common way of parsing in large pieces of text

is to use Regular Expressions (or regex for short). Regex is typically used to search within a large text via a small query string.

Information parsed is typically structured into trees (or abstract syntax trees to be precise). The leaf nodes together would end up in the original input. However, the parent nodes provide the structure needed by a computer to understand the input. The tree is based on a defined grammar and if a tree cannot be formed, a `syntax error` is generated. For example, `the cat on the hill` is fine but `cat the hill on the` is not.

Contest Information

Weblink: https://www.hackerrank.com/week-4-cse212-1718

There are 6 problems with a total of 120 points. Score 25 to get 50% of this week's grade and 60 to get 100% of this week's grade. The logistics is the same as last time, with the contest being open from Monday [1/28] 8am to Saturday [2/2] 5pm. You can submit anytime on the hackerrank platform while the contest is open. You may have to also copy your accepted solution and submit the same on LMS.

Lastly, do not share code with / copy code from one another. We are checking for plagiarism and students found to be cheating will receive **0** in their week's evaluation. As the labs are weighted at 60%, every week is weighted at approximately 5% of the total grade. The plagiarism check is based on MOSS [or Measure of Software Similarity] and that cannot be fooled by simple variable name changes. You have been warned.

Good luck, have fun!