

Assignment

January 19, 2012

Nature: Programming

Number: 2

Submission Date: 27/01/2012

Problem 1 *In this program, the attempt is to see if there are missing parantheses of any kind in a given string of symbols. The symbols include {, (, [, and their corresponding closing symbol, },), and]. An example well formed string is (({ }[])) and an example non-well formed string is ((}) { }. Write a program to take as input a space separated string of such symbols and outputs YES/NO depending on whether the string is well formed or not.*

Problem 2 *The goal of this problem is to implement the following set operations using linked lists. A set S is represented using a linked list by having one element of the set in each node. For instance, $S = \{12, 17, 9, 22\}$ is represented as head $12 \rightarrow 17 \rightarrow 9 \rightarrow 22$. The universe is the set of positive integers. Now, write programs to :*

(a) compute the union of two sets S_1 and S_2 . The output is a new list S that contains elements in $S_1 \cup S_2$.

(b) compute the intersection of two sets S_1 and S_2 . The output is a new list that contains the elements in $S_1 \cap S_2$.

Problem 3 *The following problem asks you to convert an infix expression that is given without parentheses and add parentheses to it for every subexpression. For instance, the expression $2 + 3 * 5$ will be converted to $(2 + (3 * 5))$.*

Now write a second program to take such an expression and evaluate the same without converting it into a postfix expression. The output should include the contents of the stack every k^{th} step of the program for k given as an input.

Problem 4 *Implement the following program using linked lists. Given are n integers. Create a linked list of these n integers. From this linked list, delete every fifth element (from the head). If n is not divisible by 5, stop this when $\lfloor (n/5) \rfloor$ elements are deleted. Of the remaining elements, delete every 7th element until $\lfloor (n/7) \rfloor$ elements are deleted. Keep doing so for every prime number 11, 13 etc. till the list is empty. If the length of the linked list remaining is less than the prime p that is being used presently, then count in a cyclical manner till one element is deleted. After one element is deleted, advance to the next prime. When the list has more elements than p , then*

$\lfloor (n/p) \rfloor$ elements are deleted.

At the end of this procedure, then only one element shall remain at the end. Print this element as the last line of the output file named `output.txt`. The output contains a prime number and the elements deleted using that prime number. For example, a sample output could be:

```
5 : 76, 35, 75
7 : 15, 22
11 : 83
13 : 42
...
...
93 : 26
38
```

where 38 is the last element in the linked list, and for instance, elements 15 and 22 are deleted using the prime 7.

The input shall be given as two numbers per line comma separate: the first number is the number to be inserted and the second number is the number "after" which the insert takes place. The first line of the input is a single number that is the first element to be inserted in the list. The value of n is not given as part of input, but n is the number of elements in the list.

```
12
82, 12 (That is, insert 82 after 12, so the list is  $12 \rightarrow 82$ .
15, 12 (That is the new list is  $12 \rightarrow 15 \rightarrow 82$ ).
```