Discrete Structures
CS 241 - 001
Department of Physical and Computer Sciences
Medgar Evers College

Exam 1: Midterm Project Proposals

Directions: Choose one (1) of the two (2) midterm project proposals and submit your choice on blackboard. Once approved, read the instructions carefully and complete all the tasks of the assignment.

Simple 4-Statement Propositional Calculator

Create a 4-statement propositional calculator program. The program should display a menu with the following options

1. Negation
2. Conjunction
3. Disjunction
4. Implication
5. Equivalence
6. Store
7. Quit

and then, prompt the user to select one of the chioces. If the user selects options 1 through 5, the program should prompt the user to enter the necessary operand(s). The only valid operands shound be A, B, C, D, R, 0, 1, 2, 3 and 4. These operands represents

A	T	T	T	T	T	T	T	T	F	F	F	F	F	F	F	F
B	T	T	T	T	F	F	F	F	T	T	T	T	F	F	F	F
C	T	T	F	F	T	T	F	F	T	T	F	F	T	T	F	F
D	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
R*	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
1	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
2*	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3*	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
4*	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

* **indicates the default value of the statement.**

If everything is valid, preform the selected operation with the selected operand(s), store the result in R and display R in row form enclosed in square braces; otherwise, display an error message. For example, if disjunction is selected along with the operands A and B, then R will be assigned

$$[T\ T\ T\ T\ T\ T\ T\ T\ T\ T\ T\ T\ F\ F\ F\ F]$$

and it will be displayed as above.

If the user selects option 6, the program should prompt the user to enter a memory location which is 2, 3 or 4. If one of those values is selected, store the current value of R in that statement letter, and then, display it in row form enclosed in square braces; otherwise, display an error message. Until option 7 is selected, continue to display the menu and prompt the user to make a selection.

Your program must

- have a function for each operation (conjunction, disjunction, negation, implication, equivalence).
- represent each statement letter with a boolean array.
- display truth values as T and F (case sensitive).

Once you finish with your program, use the program to generate a truth table of

$$(A' \to C') \wedge (A \to D) \wedge (B \to C) \to (B \to D)$$

that includes all intermediate wffs in order.

For extra credit, create a log file that holds a record of the all operations performed before the program terminated. It should provide the operation on its own line and the result of the operation on the following line. Furthermore, it should represent the operators conjunction, disjunction, negation, implication, equivalence, and storage with the symbols `&`, `|`, `!`, `>`, `=` and `@` respectively where storage is a unary operator. For example, suppose the user performed the task from the above example, performed a storage in statement letter 3 and then terminated the program. The log file of this run should look similar to the following

```
A | B
[T T T T T T T T T T T T F F F F]
@3
[T T T T T T T T T T T T F F F F]
```

Fundamental Theorem of Arithmetic

The fundamental theorem of arithmetic states

> Every positive integer greater than 1 can be represented as an unique (ignoring ordering) prime factorization (product of primes).

Create a program that is a prime factorization calculator for integers between 2 and 1000. The program should prompt the user to enter a positive number greater than 1, and then, it displays the prime factorization of the number in ascending order in exponential form. For example, if the user entered 56, the program should display

$$56 = (2\hat{}3)(7\hat{}1)$$

. Furthermore, the program should continue to prompt the user until the user enters 0.

Your program must

- generate the prime numbers; it cannot just read the numbers from a file.
- have at least three functions excluding the main function.
- not include prime numbers with zero powers in the prime factorization of the input of the user.

Once you finish with your program, use the program to help answer the following questions
If $\sigma(n)$ equals the sum of n's factors and $\tau(n)$ equals the count of n's factors, then

(a) if $n = p$ is prime, what is $\sigma(n)$ and $\tau(n)$?

(b) if $n = pq$ where p and q are both prime, what is $\sigma(n)$ and $\tau(n)$?

(c) if $n = p^m$ where p is prime and $m > 1$, what is $\sigma(n)$ and $\tau(n)$?

For extra credit, create a file that lists the prime factorization of all the numbers between 2 and 1000. in ascending order.