Welcome to Progfest 2010

Hosted by:

**Department of Computer Science &
Association for Computing Machinery**

**California State University
Los Angeles**

Tips:

Read each question carefully before trying to solve it.

Unless otherwise noted, all data input should come from standard IO. If a program requires an input file, it will be specified. For those of you who do not know what we mean by standard IO, it is when you prompt for input:

Example:
Input:
        Enter a number> 1
        Enter a number> 2
        Enter an operation> add

Output: 3

Good luck and have fun!

Problem #1

Shortest Path in Alabama

Write a program to find the shortest routing and distance between two Alabama cities using the following distance table. You are not allowed to use any other manually computed distances in your program.

Alabaster-Birmingham 24 miles
Alabaster-Montgomery 71 miles
Birmingham-Huntsville 103 miles
Birmingham-Tuscaloosa 59 miles
Demopolis-Mobile 141 miles
Demopolis-Montgomery 101 miles
Demopolis-Tuscaloosa 65 miles
Mobile-Montgomery 169 miles
Montgomery-Tuscaloosa 134 miles

Example 1:
    Enter city #1: Demopolis
    Enter city #2: Birmingham
    Shortest routing and distance:
    Demopolis-Tuscaloosa-Birmingham, 124 miles

Example 2:
    Enter city #1: Mobile
    Enter city #2: Huntsville
    Shortest routing and distance:
    Mobile-Montgomery-Alabaster-Birmingham-Huntsville, 367 miles

Problem #2

Prime Numbers

So far it is mathematically impossible to write a formula to generate prime-numbers exclusively. But we do know how to check for prime numbers, as prime numbers are numbers that can only be divided by themselves and by 1.

Write a program that takes an integer as input x and an output option {MAX,ALL}. Your program should calculate all of the prime numbers up to x and display the output according to the output option.

Input
>   1) An integer x, such that $1 <= x <=$ infinity
>   2) An option, MAX or ALL

>   if option==MAX
>   Output
>   The largest prime number in the sequence S ={1,2,3,...,x}
>   if option==ALL
>   Output
>   S

>   [Note: Do not count 0 and 1 as prime numbers.]

Examples
>   Input:
>   > 10 MAX

>   Output:
>   7

>   Input:
>   > 10 ALL

>   Output:
>   2 3 5 7

Problem #3

Deterministic Finite Automatons

In the theory of computation, a deterministic finite state machine—also known as deterministic finite automaton (DFA)—is a finite state machine where for each pair of state and input symbol there is one and only one transition to a next state. DFAs recognize the set of regular languages and no other languages.

A DFA will take in a string of input symbols. For each input symbol it will then transition to a state given by following a transition function. When the last input symbol has been received it will either accept or reject the string depending on whether the DFA is in an accepting state or a non-accepting state.

Write a program that simulates a DFA that accepts the language L, where

$$L = \{ w \in \{0,1\}^* \mid w \text{ ends with a } 0\}$$

This means that your machine will check the word w, a string of 0s and 1s, one index at a time and determine if the string is in the language L. Meaning that w is any string of 0s and 1s and ends with a 0.

Examples:

Input:	Output:
DFA> 011100100101	Rejected 011100100101
DFA> 0110000	Accepted 011000
DFA> 0123000	Rejected 0123000

Problem #4

Making Change

You are a manager at a local fast food restaurant. You find that your employees spent about on average 2 minutes making change per transaction. At $10/hr to employee someone, thats $0.33 per transaction you loose on making change. Given that you serve about 100 customers/day , thats $33 dollars you loose every day. You think to yourself, "Jee-wiz, can't this be automated?"

The first step, which is your challenge is given an amount, how many 10's,5's,1's,quarters, dimes, nickels, and pennies (you don't accept bills larger than 20's) does your hypothetical machine dispense to the customer ideally, minimizing the amount of items dispensed, assuming that you have an infinite supply of coins and paper bills to dispense.

Input

A currency amount to be dispensed followed by a newline.

Output

For each item 10's,5's,1's,quarters,nickels, and dimes the number to dispense followed by a space and then the item with a comma in between the items.

Sample input 1
$12.47

Sample output 1
1 10, 0 5, 2 1, 1 q, 2 d, 0 n, 2 p

Sample input 2
$9.75

Sample output 2
0 10, 1 5, 4 1, 3 q, 0 d, 0 n, 0 p

Problem #5

ROT13

According to Wikipedia, ROT13 ("rotate by 13 places") is a simple Caesar cipher used for obscuring text by replacing each letter with the letter thirteen places down the alphabet. A becomes N, B becomes O, and so on up to M, which becomes Z, then the sequence reverses: N becomes A, O becomes B, and so on to Z, which becomes M.

The algorithm is used in online forums as a means of hiding joke punchlines, puzzle solutions, movie and story spoilers, and offensive materials from the casual glance. A noticeable feature of this cipher is that it is symmetrical; that is, to undo ROT13, the same algorithm is applied, so the same code can be used for encoding and decoding.

Your program must be able to apply the ROT13 cipher to its input, and output the result only.

upper case letters A through Z need to be considered, and spaces must be left untouched, non-alpha-numeric characters can be ignored (punctuation, special characters, etc).

Example 1:
Input string: EXAMPLE ONE
Output string: RKNZCYR BAR

Example 2:
Input string: ENIVAR
Output string: RAVINE

Problem #6

Stack Machine Interpretation

Many virtual machines, e.g. the Java VM, are based on the notion of a run-time stack to hold data. This is like a stack of plates where data may only be removed from or added to the top.

Code is of the form:
        PUSH 1
        PUSH 2
        ADD
        RESULT
        => 3 (output from stack)

PUSH 1 pushes a 1, PUSH 2 pushes a 2 and ADD removes the top 2 values on the stack, adds them and pushes the result (so the 1 and 2 would be replaced by 3). RESULT is a special operation that will display the current value at the top of the stack.

Write an interpreter for such stack machine code. In addition to PUSH which will always be followed by an unsigned integer, implement ADD, SUB, MUL, DIV, and RESULT operations. The operators may be abbreviated to A, S, M, D, and R. All operands will be integers. You may assume integer division when implementing DIV.

| Example 1:
Enter virtual machine commands:
P 1
P 2
A
P 3
M
P 1
S
P 2
D
R
=> 4 | Example 2:
Enter virtual machine commands:
P 1
P 2
P 3
P 4
P 5
M
A
P 6
S
A
A
P 10
D
R
=> 2 |

Problem #7

Highway Patrol meets Modern Warfare

You live in the year 2030 and the world has become 99% automated. Highway patrol officers have been replaced by unmanned aerial vehicles (UAVs). The job of these is to collect the following information on vehicles traveling down a certain stretch of highway.

Data collected:
License plate number, Time at Checkpoint A, Time at Checkpoint B

[Note: Checkpoint A and Checkpoint B are always 5 miles apart.]
[Note: 1 mile is exactly 5280 ft.]

Currently the state employs analysts to go over the data by hand and determine who receives a ticket and who does not. We want to hurry this process up by only requiring an analyst to enter a few pieces of information into a computer program. We need you to write that program.

Ticket prices are calculated like so: base + fee x <number of mph over the speed limit>
In California in 2030 the base is $150 and the fee per mph over the speed limit is $5

Time on the UAV is kept in the number of minutes past midnight, so 10AM is 10 hours * 60 minutes = 600, 10:30AM = 630, 10PM = 60(12 + 10) = 1320

[Note: speed should be truncated, so a car moving 4.341 mph over the speed limit will be calculated at 4 mph over the speed limit.]

Your program should accept the following input:
License plate number
Time at Checkpoint A (in minutes)
Time at Checkpoint B (in minutes)
Speed limit in the zone (MPH)

We are only interested in who deserves a ticket, not those obeying the law.

Input will terminate with the end of file.

Example:

Input from file:
ABC123 1230 1235 55
DEF456 1230 1237 55

Output:
Issue ticket to ABC123 for $175

Problem #8

Computer Science is all about Matrices and Binary Numbers

You are working for a company that crunches a lot of data that comes in from various sources. All of the data that you work with is represented by matrices. The matrices come in in streams of data one at a time where each line of the matrix is delimited by a #.

For this particular project you're on, they want you to write a program that can read in a matrix a value and a command {LT,GT,EQ} where LT = Less than, GT = Greater than and EQ = equal. Your program should then look in the matrix entries and find all the entries that are LT/GT/EQ to the value input and return a binary mask of its findings, where an entry that meets the criteria is represented by a 1 and an entry that doesn't meet the criteria is represented by a 0.

Keep in mind that each entry is 4 digits long. So each entry can range from 0000 to 9999.

The output should consist of the matrix that was input and the binary mask as follows.

Example
      input> #01990057#01240075 150 LT

      output>
          Matrix
                0199 0057
                0124 0075

          LT 150
                0 1
                1 1

Perfect Number Checker

A positive integer is said to be a perfect number if it is equal to the sum of its positive divisors less than itself. For example, 28 is perfect, because

$$28 = 1 + 2 + 4 + 7 + 14$$

On the other hand, 12 is not perfect, because

12 != 1 + 2 + 3 + 4 + 6

You are to write a program that prompts the user to enter a positive integer and responds by reporting whether or not the given number is perfect.

Example 1:
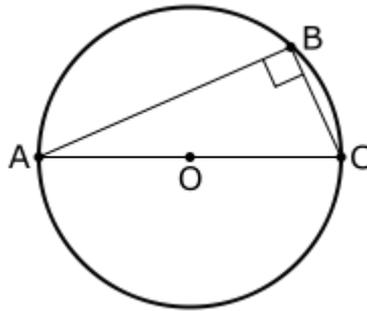Enter a positive integer: 12
12 IS NOT perfect.

Example 2:
Enter a positive integer: 28
28 IS perfect

Problem #10

Thales' Triangle

In geometry, Thales' theorem (named after Thales of Miletus, an ancient Greek philosopher) states that if A, B and C are points on a circle where the line AC is a diameter of the circle, then the angle ABC is a right angle. Thales' theorem is a special case of the inscribed angle theorem.

Write a program that can accept 3 points, ABC in the Cartesian plane (xy-plane). Your program should check the input and determine if the triangle formed by the input is a Thale's triangle.

Assume that all input re

Example1:
Input A> 2 0
Input B> -2 0
Input C> 0 2

Output> ABC is a Thales Triangle

Example2:
Input A> 10 0
Input B> 8 0
Input C> 2 2

Output> ABC is not a Thales Triangle