# 15110 PRINCIPLES OF COMPUTING – EXAM 1A – FALL 2011

| | |
|---|---|
| 1 | _____ |
| 2 | _____ |
| 3 | _____ |
| 4 | _____ |
| 5 | _____ |
| TOTAL | _____ |

Name _____ Section _____

*Directions: Answer each question neatly in the space provided. Please read each question carefully. You have 50 minutes for this exam. No electronic devices allowed. Good luck!*

1. (20 pts) This problem covers our introduction to computing.

(a) Match the computing device with its historical significance. (5 pts)

| | |
|---|---|
| _____ Colossus | A. predicted the outcome of the 1952 U.S. presidential election |
| _____ Difference Engine | B. designed to compute ballistic trajectory tables for WW2 |
| _____ ENIAC | C. used to help break the German Enigma code |
| _____ Hollerith Machine | D. designed to compute function tables for navigation |
| _____ UNIVAC | E. used to count the 1890 U.S. census using punched cards |

(b) Choose the best answer for each of the following questions. (2 pts)

_____ Where was the first personal computer with a graphical user interface created?

a. Apple b. IBM c. Kodak d. Microsoft e. Xerox

_____ Where was the World Wide Web invented?

a. Bolt, Beranek, and Newman (BBN)
b. Carnegie Mellon University (CMU)
c. Defense Advanced Research Projects Agency (DARPA)
d. European Organization for Nuclear Research (CERN)
e. National Science Foundation (NSF)

(c) Babbage sets up his function computing machine with the following values:

$\Delta^3 f(0) = 1$ $\Delta^2 f(0) = 2$ $\Delta f(0) = 3$ $f(0) = 9$

Using his machine, what would he compute for the following values? (3 pts)

$f(1) =$ _____

$f(2) =$ _____

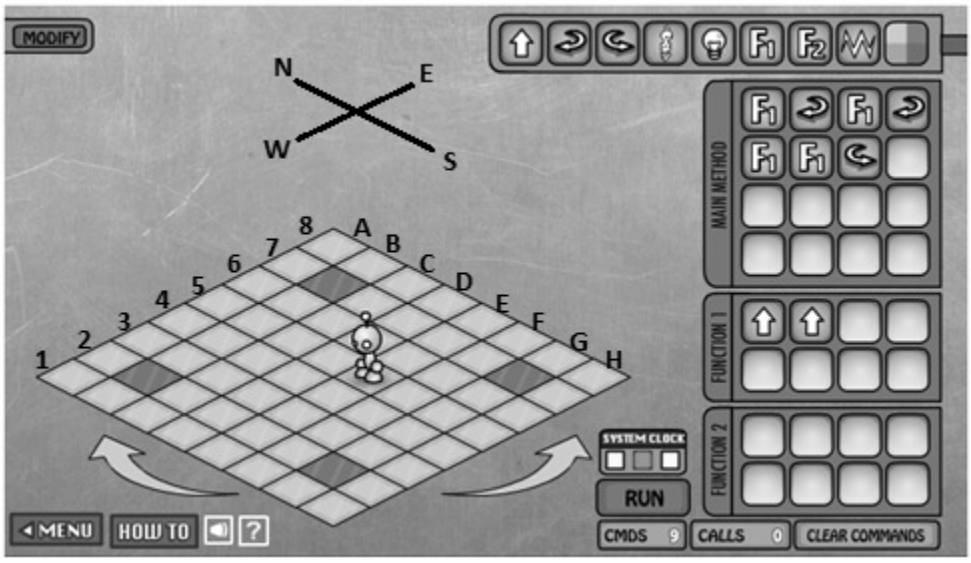
$f(3) =$ _____

(d) According to Moore's Law, the power of computing devices doubles approximately every two years. Given this observation, how many years would we have to wait for current machines to become 16 times more powerful than they are today? (2 pts) _____

(e) A computer has a hard drive of 256 GB. Compute the number of bytes as a power of 2. (2 pts) ____ 2 bytes

(f) In the book Blown To Bits, you read about how Kodak missed the exponential growth in digital technology which resulted in a loss of jobs and money for the company. In one sentence, state how exponential growth in the use of digital technology will likely cause similar problems for the United States Postal Service. (3 pts)

(g) LightBot is facing west (W) in the world below. If the robot follows the given program, give the final resting place of the robot (row and column) and its direction (N, S, E or W). Note: None of the instructions depends on color. (3 pts)




Final resting place: Row (A-H): _____ Column (1-8):_____ Direction (N/S/E/W):_____

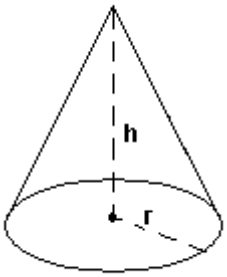
2. (20 pts) This problem focuses on the basics of writing simple Ruby expressions and functions.

(a) For each of the following Ruby expression, write down the value that would be output if the expression was evaluated in `irb`. (5 pts)

```
5 + 4 * 3 - 2          _____
79 % 10                _____
96 / 10                _____
3 ** 2 * 2             _____
1 / 2  * 5             _____
```

(b) Write a Ruby function `cone_volume` that has two parameters for the `height` of a cone and the `radius` of the cone's base and **returns** the volume of the cone given by the formula shown to the right. (3 pts)

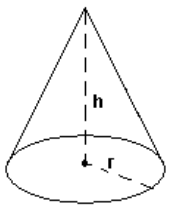
HINT: Recall that π is represented in Ruby as `Math::PI`



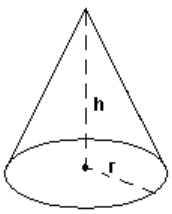

$$V = \frac{1}{3}\pi r^2 h$$

(c)Using your answer to part (b), write a Ruby instruction to print the total volume of the cones shown below. (3 pts)

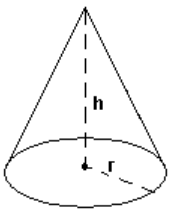
_____




h = 10, r = 4




h = 5, r = 6




h = 7, r = 2

(d) Consider the following Ruby function where n is assumed to be a positive integer:

```
def mystery(n)
      j = 0
      k = 1
      for i in 1..n do
            j = j + k
            k = k + 2
      end
      print j
end
```

| i | j | k |
|---|---|---|
| --- | 0 | 1 |
| 1 | _____ | _____ |
| 2 | _____ | _____ |
| 3 | _____ | _____ |
| 4 | _____ | _____ |
| 5 | _____ | _____ |

Trace this function for n = 5, showing the value of j and k in the table above after each iteration of the loop. The initial values of j and k are given for you in the table. (5 pts)

(e) State what mathematical function `mystery` is computing in terms of `n`. (2 pts)

_____

(f) Using the function in part (d), briefly explain what is wrong with the Ruby computation below. (2 pts)

```
x = mystery(5) * mystery(7) * mystery(8)
```

3. The following question deals with arrays of elements in Ruby.

(a) Assume the following list definition in Ruby:

```
fruit = ["apple", "cherry", "peach", "grape", "banana", "orange", "melon"]
```

What would displayed in `irb` for each of the following Ruby expressions? (12 pts)

```
fruit.first                                    _____________________________

fruit.length                                   _____________________________

fruit[2]                                       _____________________________

fruit.include?("PEACH")                        _____________________________

fruit.each { |item| print item + " " if item.length == 6 }

     _____________________________________________________________________

fruit.delete_if { |item| item < "fig" }

     _____________________________________________________________________
```

(b) Recall the revised algorithm to return a list of all primes between 2 and n:

```
def sieve(n)
     numlist = Array(2..n)
     primes = []
     while numlist.first < Math.sqrt(n) do
          primes << numlist.first
          numlist.delete_if { |x| x % primes.last  == 0 }
     end
     return primes + numlist
end
```

Trace the algorithm for n = 10, showing the contents of the arrays `numlist` and `primes` at the end of each iteration. The initial values are given for you. You may not need all of the blanks shown below. (8 pts)

| numlist | primes |
|---|---|
| [2,3,4,5,6,7,8,9,10] | [] |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |

4. (20 pts) We wish to define a Ruby function `get_minimum` to return the minimum in an array of integers. For example, `get_minimum([5,3,6,7,2,8,4,3])` should return `2`.

(a) Complete the following iterative function for `get_minimum` assuming the list is never empty. (8 pts)

```
def get_minimum(list)

      min_so_far = _____________________

      for i in (_____________________ .. _____________________) do

            if _____________________ < min_so_far then

                  _____________________________________________

            end
      end

      return _____________________________
end
```

(b) Consider the following recursive algorithm for returning the minimum in an array of integers:

1. If the list has only one element, then return that element as the minimum.
2. Otherwise do the following:
    a. Compute the minimum of the list that does not include the first element.
    b. If the first element is smaller than the minimum computed in step a,
        return the first element. Otherwise, return the minimum computed in step a.

Complete the following recursive function for `get_minimum` assuming the list is never empty. (8 pts)

RECALL: `a[i..j]` evaluates to a new array containing the $i^{th}$ through $j^{th}$ elements of array `a`

```
def get_minimum(list)

      return ____________________ if ____________________________________

      min_without_first_element = get_minimum(_____________________________)

      if ____________________________________________________________ then

            return list.first
      else
            return min_without_first_element
      end
end
```

(c) Briefly state why the function in part (b) is *recursive*. (4 pts)

_____

5. This question deals with searching and sorting algorithms.

(a) Consider a new implementation of linear search that examines the elements of a list in reverse order, from last to first, looking for some key. Complete the following Ruby code for this new function. (6 pts)

```
def reverse_search(list, key)

      i = _____________________              # index of last element

      while _____________________________ do

            return i if list[i] == key
            i = i - 1
      end

      return ___________________

end
```

(b) Let n represent the length of the list. Using big O notation, what is the worst case order of complexity for the algorithm above as a function of n? (2 pts) _____

(c) TRUE or FALSE: If the `list` is sorted for `reverse_search`, the worst case order of complexity (in terms of big O) improves. (2 pts) _____

(d) Consider the following algorithm for sorting a Ruby-based list (a.k.a. "Selection Sort"):

1. Set i = 0
2. While i < list.length-1 do the following:
    a. Set j equal to the index of the minimum element in the list
        from index i to the end of the list.
    b. Swap (exchange) the elements at index i and index j in the list.
    c. Add 1 to i.

What are the contents of the list after each iteration of the loop if list = [8, 3, 7, 4, 5] ? (8 pts)

```
Initial list:              [8, 3, 7, 4, 5]

After first iteration:     ___________________________________

After second iteration:    ___________________________________

After third iteration:     ___________________________________

After fourth iteration:    ___________________________________
```

(e) (2 pts) Circle the best answer: In terms of worst case order of complexity (using big O), this sort is

a. better than insertion sort  b. worse than insertion sort  c. the same as insertion sort