

# Computer Science Foundation Exam

March 9, 2001

Section I A

**No Calculators!**

Name: \_\_\_\_\_

SSN: \_\_\_\_\_

In this section of the exam, there are three (3) problems.

You must do all of them.

The weight of each problem in this section is indicated with the problem.

The algorithms in this exam are written in a combination of pseudocode and programming language notation.

Partial credit can not be given unless all work is shown.

As always, be complete, yet concise, and above all be neat, credit can not be given when your results are unreadable.

(1, 20%) Given the following array of integers and algorithm, answer the questions below. Assume that the global array **X[1..n]** is correctly declared and contains the values shown.

<b>Array X</b>	<b>3</b>	<b>1</b>	<b>5</b>	<b>2</b>	<b>4</b>	<b>2</b>
position	1	2	3	4	5	6

```

algorithm P1
  a, b, i, j, k : integer

  a ← 0
  b ← 0
  i ← 1
  j ← 6
  loop
    if (X[i] <= X[j]) then
      a ← a + X[i]
      k ← j
      loop
        exitif (k = i)
        X[i] ← X[i] + X[k]
        k ← k - 1
      endloop
      i ← i + 1
    else
      b ← b + X[j]
      k ← i
      loop
        exitif (k = j)
        X[j] ← X[j] + X[k]
        k ← k + 1
      endloop
      j ← j - 1
    endif
    exitif (i > j)
  endloop
endalgorithm

```

a) Show the array **x** after the procedure has completed execution (2pts each)?

<b>Array X</b>						
position	1	2	3	4	5	6

b) What value will the variables **a** and **b** contain when the algorithm is finished (4pts each)?

<b>a</b>		<b>b</b>	
----------	--	----------	--

(2, 14%) In the following Postfix expressions all values are single decimal digits and the operations are addition "+", subtraction "-", multiplication "\*" and division "/". In each box below the Postfix expression in part a), show ONLY the contents of the stack at the indicated point in the Postfix string (point A, B or C). Put the final answer in the blank. If the Postfix string is invalid, carry the operations as far as possible and write "invalid" as the answer.

a)  $7\ 2\ 4\ -\ 3^A\ *\ 5\ +\ 8^B\ 4\ /\ * 6^C\ +\ - = \underline{\hspace{2cm}}$


**A**


**B**


**C**

b) Assume that you have a stack and a queue and the standard stack and queue operations: push, enqueue, top, dequeue and pop.

Execute, in top-to-bottom order, the operations below and show the output produced by the print statements.

```

push(6)
push(4)
enqueue(pop)
push(3)
enqueue(7)
push(2)
print(pop)
push(dequeue)
enqueue(1)
enqueue(pop)
print(dequeue)
enqueue(8)
print(top)
push(dequeue)
print(dequeue)

```

output from print statements:

first output	second output	third output	fourth output

**(3, 20%)** Answer each of the following "timing" questions concerning an algorithm of a particular order and a data set of a particular size. Assume that the run time is affected only by the size of the data set and not its composition.

a) For an  $O(n!)$  algorithm, one data set with  $n = 4$  takes **72** seconds.

How long will it take for a data set with  $n = 5$ ?

\_\_\_\_\_

b) For an  $O(2^n)$  algorithm, one data set with  $n = 7$  takes **96** seconds.

If you used a different-sized data set and it took **12** seconds, how large must that data set be?

\_\_\_\_\_

c) For an  $O(n^3)$  algorithm, a friend tells you that it took **192** seconds to run on her data set. You run the same program, on the same machine, and your data set with  $n = 6$  takes **648** seconds.

What size was her data set?

\_\_\_\_\_

Given the following pseudocode segment, answer the questions below for an arbitrary  $n$ :

```
x ← 0
for i ← 1 to (n+2) do
  for j ← 1 to (2*n) do
    x ← x + j
```

d) What is the Order of this pseudocode segment?

\_\_\_\_\_

e) What will be the value of  $x$  when the **for** loops end?

\_\_\_\_\_

# Computer Science Foundation Exam

March 9, 2001

Section I B

**No Calculators!**

Name: \_\_\_\_\_

SSN: \_\_\_\_\_

In this section of the exam, there are three (3) problems.

**You must do all of them.**

The weight of each problem in this section is indicated with the problem.

The algorithms in this exam are written in a combination of pseudocode and programming language notation. Any algorithms that you are asked to produce should use a syntax that is clear and unambiguous.

Partial credit can not be given unless all work is shown.

As always, be complete, yet concise, and above all be neat, credit can not be given when your results are unreadable.

**(4, 10%)** Write a **recursive** function, called **prob4**, that will correctly print the index positions of all the occurrences of a specified character **c**, within the first **m** locations of an array **X**. You may assume that **X** is a global array which includes locations that range from **1** to **n** and is already populated with characters. Assume that **m**  $\leq$  **n**. The initial call is **prob4(c, m)**. You may use pseudocode, C, Java or Pascal syntax but points will be deducted if your meaning is not clear.

**procedure prob4( c : char; m : integer)**

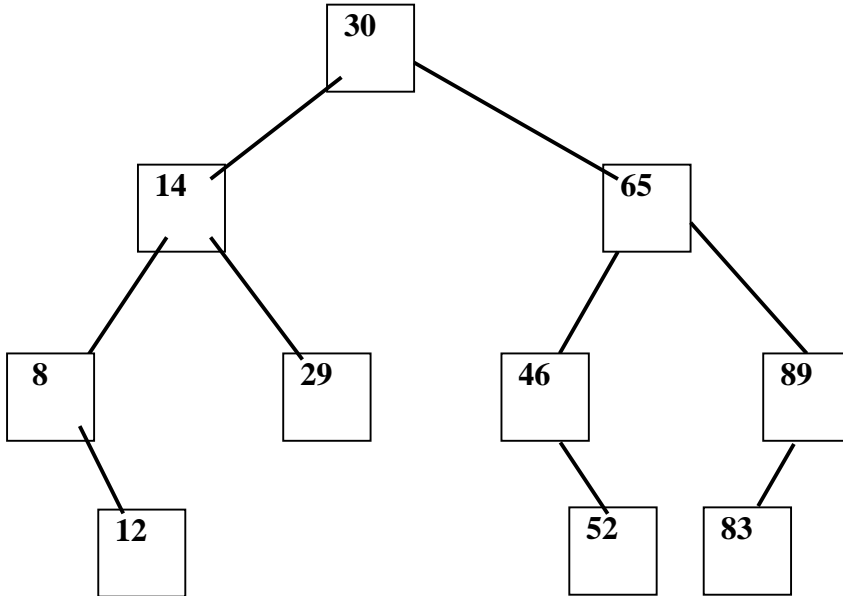
**(5, 18%)** Find the closed form or exact value for the following:  
( *n* is an arbitrary positive integer):

a)  $\sum_{i=0}^{47} (3i + 1) = \underline{\hspace{2cm}}$

b)  $\sum_{i=1}^{2k-2} (2i + 5) = \underline{\hspace{2cm}}$

c)  $\sum_{i=25}^{75} (4i - 3) = \underline{\hspace{2cm}}$

(6, 18%) Given the following Binary Tree, answer the questions below :



a) Is this a valid Binary Search Tree? (circle one) **Yes** **No**

b) List the nodes of this tree in the order that they are visited in a **preorder** traversal:

--	--	--	--	--	--	--	--	--	--

first node visited
last node visited

c) Perform the following procedure on the tree above, listing the output in the spaces below and leaving any unused spaces blank. Assume that the procedure is initially called with **P6(root, 30)** and that the tree nodes and pointers are defined as:

```

tree_node definesa record
  data isotype Num
  left, right isotype ptr toa tree_node
endrecord
tree_ptr isotype ptr toa tree_node

procedure P6(node_ptr isotype in tree_ptr, key isotype in Num)
  if (node_ptr <> NULL) then
    P6(node_ptr^.left, (key DIV 2))
    P6(node_ptr^.right, (key * 2))
    if (node_ptr^.data < key)
      print(node_ptr^.data)
    endif
  endif
endprocedure
  
```

\_\_\_\_\_