Computer Science Foundation Exam

December 14, 2000

Section I A

Solutions

(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. Assume that the procedure was called with **`P1(1, 6, 5).`**

Array X	**8**	**6**	**2**	**7**	**4**	**5**
position	1	2	3	4	5	6

```
procedure P1(i, j, k : integer)
  a, b, c, d : integer

  a ← 0
  b ← 0
  c ← 0
  d ← 0

  loop
      if (X[i] > X[j]) then
          a ← a + (X[i] - X[j])
          X[i] ← X[i] - k
          c ← c + X[i]
          i ← i + 1
      else if (X[j] > X[i]) then
          b ← b + (X[j] - X[i])
          X[j] ← X[j] + k
          d ← d + X[j]
          j ← j - 1
      else
          c ← c + k
          d ← d - k
          i ← i + 1
          j ← j - 1
      endif
      exitif (i > 5) OR (j = 1)
  endloop
endprocedure
```

a) Show the array **X** after the procedure has completed execution?

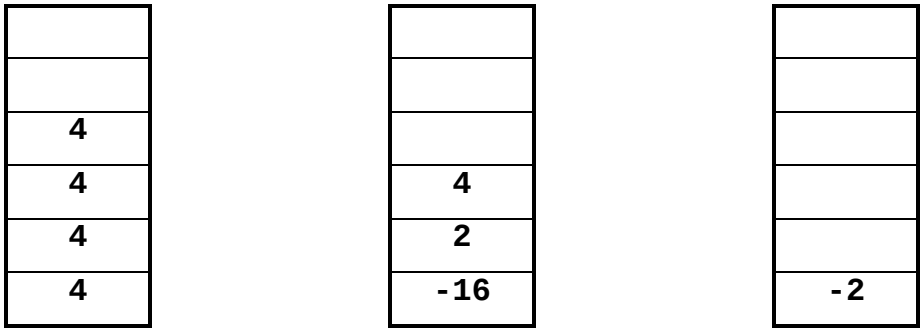
Array X	**3**	**1**	**2**	**7**	**4**	**10**
position	1	2	3	4	5	6

b) What value will the following variables contain after the **`loop`** is finished?

a	**23**	**b**	**10**	**c**	**20**	**d**	**26**

(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) **4 4 4 4 A + - * 2 4 B + + 5 / C 3 * = -6**



b) given the following stack and queue operations, taken in top to bottom order, show the output produced by the print statements:

```
push(5)
enqueue(4)
enqueue(2)
push(7)
enqueue(pop)
push(dequeue)
push(3)
print(dequeue)
print(pop)
push(8)
enqueue(6)
enqueue(pop)
push(dequeue)
print(pop)
print(dequeue)
```

output from print statements:

2	3	4	7
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 $\mathbf{O(n^2)}$ algorithm, one data set with **n = 7** takes **98** seconds.

How long will it take for a data set with **n = 5**? **50 sec.**

b) For an $\mathbf{O(2^n)}$ algorithm, one data set with **n = 4** takes **15** seconds.

If you used a different-sized data set and it took **60** seconds, how large must that data set be? **n = 6**

c) For an $\mathbf{O(\log_2 n)}$ algorithm, a friend tells you that it took **12** seconds to run on her data set. You run the same program, on the same machine, and your data set with **n = 32** takes **20** seconds.

What size was her data set? **n = 8**

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

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

d) What is the Order of this pseudocode segment? $\mathbf{O(n^2)}$

e) What will be the value of **x** when the **for** loops end? $\frac{n(n+1)}{2}$

(4, 10%) Assume that a global array of characters, called **X**, exists and includes locations that range from **1** to **n**. In the space below, write a **recursive** procedure, called **Reverse**, that exactly reverses the order of the characters in the array.
You may assume that the array is already populated with alphabetic characters before the procedure is initially called and you should only write the code contained in **Reverse**.
You may also assume that the procedure will initially be called as follows:
Reverse(1, n). You may use pseudocode or C or Pascal syntax but points will be taken off if your meaning is not clear.

```
procedure Reverse(i, j : integer)
     temp : integer;

     if (i < j)  {
        temp ← X[i];
        X[i] ← X[j];
        X[j] ← temp;
        Reverse( i+1, j-1 )
     }
}
```

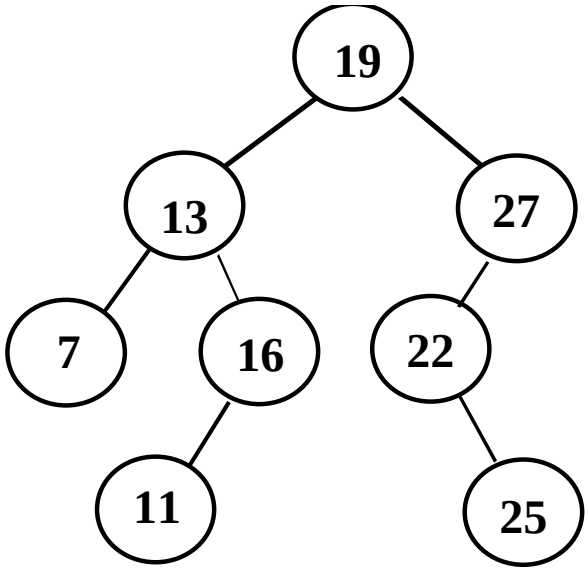
(5, 18%) Find the closed form or exact value for the following:
(n is an arbitrary positive integer):

a) $\sum_{i=0}^{60}(5i-3)$ = **8967**

b) $\sum_{i=1}^{2k+1}(4i+1)$ = $\mathbf{8k^2 + 14k + 5}$

c) $\sum_{i=40}^{90}(3i-2)$ = **9843**

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



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

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

19 13 7 16 11 27 22 25

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)** and that the tree nodes and pointers are defined as:

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

procedure P6(node_ptr isoftype in tree_ptr)
   if (node_ptr <> NULL) then
         P6(node_ptr^.right)
         print(node_ptr^.data)
         P6(node_ptr^.left)
      endif
   endif
endprocedure
```

27 25 22 19 16 11 13 7