

Computer Science Foundation Exam

May 3, 2000

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.

If you need extra room to do work to be graded then do so on the last page attached to this test. Make sure to clearly label the problem you are working on.

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 numbers and procedure, answer the questions below. Assume that the global array $X[1..n]$ is correctly declared and contains the values shown. (So $n=8$.) Assume that all division(/) is integer division. So, for example, $17/3$ evaluates to 5.

Array X	0	0	0	0	0	0	0	0
Position	1	2	3	4	5	6	7	8

```

procedure ArrayOperation(i, j: integer)
  mid, k : integer;
  mid ← (i + j)/2;
  for k ← i to j do
    if (k MOD 2 = 0) then
      X[k] ← X[k] + i
    else
      X[k] ← X[k] + j
    endif
  endfor
  if (mid > i) then
    ArrayOperation(i, mid);
    ArrayOperation(mid + 1, j);
  endif
endprocedure

```

a) (8 pts) Show the array x after the procedure call $\text{ArrayOperation}(1,8)$ has executed?

Array X	14	3	16	5	22	11	24	13
Position	1	2	3	4	5	6	7	8

b) (6 pts) Consider the case where each element of the array X is equal to m , where m is a positive integer. Now consider executing the procedure $\text{ArrayOperation}(1,8)$. What will the sum of the elements of the array X be, in terms of m , after the procedure call?

Here, the values in the array will be $m+14$, $m+3$, $m+16$, $m+5$, $m+22$, $m+11$, $m+24$, and $m+13$. These add up to $8m+108$.

c) (6 pts) Consider the case where the size of the array X is 2^d , where d is a positive integer, and each entry of the array is initialized to 0. After executing the procedure call $\text{ArrayOperation}(1,\text{size})$, where size is equal to 2^d , what index of the array will store the maximal value of the array? What is the value that will be stored in this index?

The index is the $2^d - 1$ index, and the value stored there will be $d \cdot 2^d$.

(2, 18%) The following are Postfix expressions. All values are single decimal digits and the operations are addition "+", subtraction "-", multiplication "*" and division "/". (Note that the final answer and intermediate answers in the stack may not be single decimal digits.) In each box below the Postfix expression, 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. (6 points each)

a) $6\ 7\ 9\ +\ \mathbf{A}\ -\ 2\ /\ \mathbf{B}\ 11\ 2\ 2\ /\ +\ \mathbf{C}\ * = 60$

16
6

A

-5

B

12
-5

C

b) $2\ 8\ -\ 12\ *\ \mathbf{A}\ 2\ 5\ 6\ +\ \mathbf{B}\ -\ /\ 3\ 3\ \mathbf{C}\ /\ * = 8$

-72

A

11
2
-72

B

3
3
8

C

Next to each Postfix expression, circle one answer to indicate if it is a valid Postfix string or not: (no extra credit for providing the answer, if it is valid) (3 points each)

c) $3\ 5\ 7\ -\ +\ 3\ 2\ 9\ *\ /\ -$ **Valid**

d) $8\ 2\ +\ 8\ 2\ *\ 2\ 3\ -\ 3\ *\ /\ +$ **Valid**

(3, 20%) Answer each of the following "timing" questions concerning an algorithm of a particular order and a data instance of a particular size. Assume that the run time is affected by the size of the data instance and not its composition. (Assume that $\lg n = \log_2 n$.)

a) (4 pts) For an $O\left(\frac{n}{\lg n}\right)$ algorithm, an instance with $n=32$ runs takes 96 milliseconds.

How long would it take to run the algorithm with $n=64$? **160ms**

b) (4 pts) For an $O(n^4)$ algorithm, an instance with $n = 20$ takes **256** milliseconds.

If you used a different-sized data instance and it took **81** milliseconds how large must that instance be? **15ms**

c) (4 pts) For an $O(k^n)$ algorithm, where k is a positive integer, a friend tells you that her instance of size m took **2** seconds to run. You run an instance of size $m+3$ and find that it takes **54** seconds to run. What is the value of k ? **3**

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

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

d) (2 pts) What is the Order of this code segment, in terms of n ?

$O(n^2)$

e) (6 pts) What will be the value of x (in terms of n)
when the **for** loops end?

$(3n^2 - n)/2$

Extra Work Page - Please clearly label any work on this page that you would like graded.

Computer Science Foundation Exam

May 3, 2000

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. The 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.

If you need extra room to do work to be graded then do so on the last page attached to this test. Make sure to clearly label the problem you are working on.

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

(4, 12%) Given a global array of numbers $X[1..n]$, where each element in the array is either a 0 or a 1, write a function **Find_N_Small_Val(i : integer)** that will return the i th smallest value stored in the array X . Assume that the **Find_N_Small_Val** function will be called as shown below. (For example, if the value of i passed in is 10, then answer should equal the value of the tenth smallest number stored in the array X .) Also assume that the value i passed to the function will be an integer in between 1 and n , inclusive.

$\text{answer} \leftarrow \text{Find_N_Small_Val}(i)$

In the space below, write the function **Find_N_Small_Val**.

```
Find_N_Small_Val(i : integer)

    int num_zeros, j
    num_zeros ← 0
    for j ← 1 to n do
        if (X[j] = 0) then
            num_zeros ← num_zeros + 1
        endif
    endfor

    if (num_zeros >= i) then
        return 0;
    else
        return 1;
    endif
endfunction
```


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

a) (6 pts) $\sum_{i=1}^{n+3} \left(\frac{6i+5}{n+3} \right) = 3n+17$

b) (6 pts) $\sum_{i=4}^{53} (2i-7) = 2500$

c) (6 pts) $\sum_{i=1}^{2n} (n-4i) = -6n^2 - 4n$

(6, 12%)

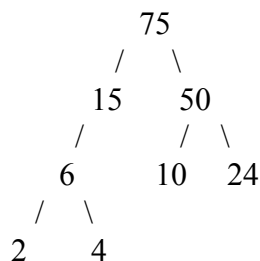
a) (8 pts) Consider the following record type definition for a binary tree node:

```
Tree_Node defines a record
  data isotype Num
  left_child isotype Ptr to a Tree_Node
  right_child isotype Ptr to a Tree_Node
endrecord
```

Now, consider the following procedure that traverses a binary tree and prints out numbers. (Note that the parameter `current_ptr` is passed in the procedure by value, but the parameter `x` is passed in by reference.)

```
Procedure Bin_Tree_Trav (current_ptr isotype in Ptr to a Tree_Node, x isotype in/out Num)
  if (current_ptr <> NIL) then
    Bin_Tree_Trav(current_ptr^.left_child, x)
    Bin_Tree_Trav(current_ptr^.right_child, x)
    if (x < current_ptr^.data ) then
      x ← x + current_ptr^.data
      print(x)
    else
      print(current_ptr^.data)
    endif
  endif
endprocedure
```

Assume that `head` is a `Ptr to a Tree_Node`. More specifically, `head` is pointing to the head of the tree below.



What would the procedure call `Bin_Tree_Trav(head,x)` print out if `x` is set to 0 before the procedure call?

2, 6, 6, 21, 10, 45, 95, 75

b) (4 pts) If we had the exact same tree as above, except swapped the 2 with the 75 inside of the tree, what would get printed out by the procedure call `Bin_Tree_Trav(head,x)`? (`x=0` before the call.)

75, 4, 6, 15, 10, 24, 50, 2

Extra Work Page - Please clearly label any work on this page that you would like graded.