Print:
First Name:............................ Last Name:............................

Login ID:...

Student Number:...

University of Toronto
Faculty of Applied Science and Engineering

Midterm Examination – October 1996

APS105F – Computer Fundamentals

Examiner – Paul Chow

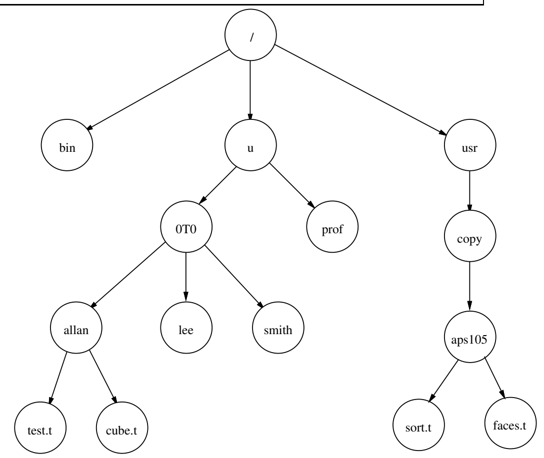
1. There are 6 questions and **10** pages. Do **all** questions. The total number of marks is 100.
2. **ALL WORK IS TO BE DONE ON THESE SHEETS!** Use the back of the pages if you need more space. Be sure to indicate clearly if your work continues elsewhere.
3. Please put your final solution in the box if one is provided.
4. **No calculators or other computing devices allowed.**
5. **Paper Type A** – No Aids permitted.

1 [15]	
2 [20]	
3 [20]	
4 [15]	
5 [10]	
6 [20]	
Total [100]	

[5 marks] 1. (a) The Unix directory tree shown below gives a part of the structure of the ECF file system. Assume that you are user **smith** and you have just logged onto your account. By adding to the figure, show the structure of the file system after you execute the following commands:

```
mkdir assignments
cd assignments
cp /usr/copy/aps105/sort.t mysort.t
mkdir ~/stuff
cp /usr/copy/aps105/faces.t ~/stuff/myfaces.t
cd ..
mv ~/stuff/myfaces.t ~/assignments/newfaces.t
rmdir ~/stuff
pwd
```

What is the output of the last command? Hint: It starts with '/'.



[2 marks] (b) Unix uses a one-way cipher for encrypting passwords. What does it mean for a cipher to be *one-way*?

After you type in your password, what does the system do before you are allowed to login?

[3 marks] (c) What are the three basic types of interfaces used by a computer to communicate with the outside world? Give an example of each.

[3 marks] (d) List three data types in Turing.

[2 marks] (e) What is the difference between *semantics* and *syntax*?

2. Consider the following fragment of a program:

```
procedure P(x : int)
    ...
end P

procedure Q(var y : int)
    ...
end Q

var A,B,C : int

A := 5
B := 7
P(A)
Q(B)
Q(B-A)
P(A+B)
Q(P)
P(12)
Q(10)
```

[6 marks] (a) Explain the mechanics of how *var* and *non var* parameters are handled. Show all of the relevant memory locations and indicate when they are created and/or destroyed. Use the calls `P(A)` and `Q(B)` in the above program as examples.

[4 marks] (b) In the program above, which procedure calls are not valid in Turing? Why?

[10 marks] (c) For the following program, draw lines to indicate the scope of all variables.

```
var A : int
var B : int
var D : int

procedure P (A : int, var C : int)
    var tmp : int

    B := B + 30
    if A < B then
        tmp := A
    else
        tmp := B
    end if

    C := tmp
end P

A := 5
B := 42
P (A, D)
put A, ",", B, ",", D
```

What is the final value of A? | |

What is the final value of B? | |

What is the final value of D? | |

[20 marks] 3. Write a pseudo code program (using a Turing-like language) that can be used to help someone convert between the Celsius and Fahrenheit temperature scales. Your program should prompt the user for input and output the temperature in the other scale. You can assume the input will be of the form:

```
0 F
17 C
```

where there is an integer followed by a letter, which will be an `F` or a `C` for Fahrenheit and Celsius, respectively. The number and letter will be separated by a space. The program ends when the letter is not `F` or `C`. Note that $(F - 32) * \frac{5}{9} = C$.

4. Assume that x and N are variables that have values assigned to them.

[5 marks] (a) Translate the following `for` loop into a semantically equivalent `while` loop.

```
for i : 1 .. 12
    x := x + 1
end for
```

[5 marks] (b) Translate the following `for` loop into a semantically equivalent `repeat/until` loop.

```
for decreasing j : N .. 1 by 2
  x := x + 1
end for
```

[5 marks] (c) Translate the following `for` loop into a semantically equivalent Turing `loop/end loop` loop.

```
for i : 1 .. N by 7
  x := x + 1
end for
```


[10 marks] 5. The algorithm for multiplying two $N \times N$ matrices is shown below.

```
for i : 1 .. N
    for j : 1 .. N
        sum := 0
        for k : 1 .. N
            sum := sum + A(i,k) * B(k,j)
        end for
        C(i,j) := sum
    end for
end for
```

What is the computational complexity of this algorithm using the "O" notation?

Explain/show how you derived your answer.

[5 marks] 6. (a) When is it appropriate to use a function and when should you use a procedure?

[10 marks] (b) Using pseudo code, write a recursive function, `FindSum`, that calculates the sum of successive integers starting at 1 and ending at N. For example:
`FindSum(N)` returns the sum of `1 + 2 + ... + N`.

Question 6 continued...

[5 marks] (c) Write an iterative version of `FindSum`.