

CS 2073, Computer Programming with Engineering Applications
Spring 1992
Exam 1

- (20) 1. Write a program segment which will read in two integers A and B and will write out the *larger* of the two numbers (or the common value if they are equal). (No declarations or comments needed.)
- (20) 2. Each of the four examples below contains a significant error. In each case say what the error is and whether it is a compile-time error (produces a message at compile time), a link-time error (produces a message at link time), a run-time error (produces a message during the run of the program, along with the program output), or a logic error (produces no error message).

(a) `N := 0; X := 13.0/N;`

(b) `if X = 0 then Y := 1;
 else Y := 2;`

(c) `porgram XYZ (input, output);`

(d) `sum := 0; count := 0;
while not eof(infil) do
 readln(infil, score);
 sum := sum + score;
 count := count + 1;`

`average := sum/count;`

- (20) 3. Consider the following code segment, where X, Y, and TEMP are declared integer.

```
X := 3; Y := 5;  
while X < Y do  
begin  
    TEMP := X;  
    X := Y;  
    Y := TEMP  
end;  
writeln(X, Y);
```

- (a) How many times will the statements inside the while loop be executed?
- (b) What is written out ?

(20) 4. Consider the following Pascal program, including a function definition:

```
program right_triang;
var A, B, C: real;

function hypot (X, Y: real): real;
begin
  if (X < 0.0) or (Y < 0.0) then
    hypot := 0.0
  else
    hypot := sqrt(sqr(X) + sqr(Y))
end; (* hypot *)

begin
  A := 3.0; B := 4.0;
  C := hypot(A, B);
  writeln(C)
end.
```

- (a) What will be printed out?
 - (b) How have we referred to the variables A and B in the function call `hypot(A, B)`? How have we referred to the variables X and Y in the function definition?
 - (c) Describe in some detail how the numbers 3.0 and 4.0 are moved around as the program is executed and how C acquires a value.
 - (d) Are the parentheses around `X < 0.0` and around `Y < 0.0` required?
- (20) 5. Write a program segment that will read integer numbers from the terminal until a *negative number* is entered. The program should find and print out the average of the numbers read in up to the negative number. The average should be printed out using *three* digits to the right of the decimal place. (Thus if you type in 80, 100, 90, and -1 on separate lines, your program should print 90.000 as the average. No declarations or comments are needed.)