void getAB(int *s, int *e)
{
    // Get a valid value for *s
    ...
    do
    {
        printf("Enter ending value (>= %d): ", *s);
        scanf("%d", e);
        if(*s > *e)
        {
            printf("Enter value >= %d!!\n", *s);
        }
    }while(*s > *e);
}

What happens if the *?s are omitted?
• gcc issues warnings; a.out crashes with “segmentation fault” error
My pseudocode

get nrows
for (row = 1; row <= nrows; row++)
    print "12...(row)
"
My pseudocode, II

get nrows
for (row = 1; row <= nrows; row++)
    // print “12...(row)\n”
    for (col = 1; col <= row; col++)
        print col
    print “\n”
Recursion--*omitted*
Too Much Data...

e.g., entering input data for testing again and again

As you correct logical errors, why is it important to test your program with the same data sets? note rephrasing
To ensure that you are isolating, addressing, and resolving one error under the same conditions.
int calcFactorial(int n) {
    int nFact = 1;
    while (n > 0) {
        nFact *= n;
        printf("%d ", nFact);
        n--
    }
    return nFact;
}

What is printed for `calcFactorial(4)`?

A. 1 2 3 4
B. 4 12 24 24
C. 4 3 2 1
D. 1 2 6 24
E. 24 6 2 1
for (row = 1; row <= 6; row++)
    for (__________________; col++)
        print col
    print "\n"

Fill in the blanks to get output on right.
A. col = 1; col <= 6
B. col = row; col <= row
C. col = 1; col <= row  1 pt
D. col = row; col <= 6
The UNIX I/O redirection symbols for input, output, and append are, respectively,
A. `<  >  >>`  
B. `<  >  >&`  
C. `>  <  <<`  
D. `>  <  <&`
What happens if we enter
   a.out < data2 > results
and results exists?
A. the program is halted with an error message
B. the output of a.out goes to the terminal
C. the output of a.out is appended to results
D. the output of a.out overwrites results  1 pt