Homework 4...

Assignment Preparation

Due date: Tuesday, February 22, bring to the review session

Testing. All `C` code in the assignment has been tested. I suggest that you do the following, when solving the problems:

- Try solving the problem on paper. Work out the details, simulate the run of the program, converge on results.
- Type the program up, compile, debug as necessary, see it run, observe results.
- If the results do not match your answers, try to figure out where you made a mistake, and **why** the output is what it is.

Name: _________________________

Section: _________________________

Problem 1.

Write a code fragment (a `while` loop or a `do-while` loop with any extra statements necessary) that will read one after another characters from standard input until the character read is `'E'` or `'e'`. After the loop is over, the code shall print the number of characters (including the final `'e'/'E'` read. Your code shall contain variable declarations, but no `main()` or `#include` declarations are necessary.

Problem 2.

Consider the following code:

```
int a[10] = {9, 8, 7, 6, 4, 5, 0, 1, 2, 3};
int i,count;

count=0;
for (i=0; i<10; i++) {
   if (i == a[i]) {count++;}
}
printf("%d\n",count);
```

What number will be printed?

Problem 3.

Write a function `int count(int list[], int size, int sign)` which takes as input an array `list` of integers and an integer number `size` specifying the size of the array, and an integer number `sign`, which takes values of 0, -1 or 1. If `sign` is 0, the function returns the number of occurences of zeroes in the array `list`. If `sign` is -1, the function returns the number of occurrences of (strictly) negative numbers in the array `list`. Finally, if `sign` is 1, the function returns the number of occurrences of (strictly) positive numbers in the array `list`.

Problem 4.

For each code fragment below determine how many times will the loop repeat, and what will be printed out.

1. Fragment 1.

```
success = 0; x=0;
while (!success) {
   if (!(x%2)) {x++;}             Number of iterations: __________
   if (!(x%3)) {x++;}
   if (!(x%4)) {success = 1;}     Output:
   x++;
   printf("%d\n", x);
}
```

2. Fragment 2.

```
for(j=6;j>0;j--) {                  Number of iterations: __________
   printf("%%");
}                                   Output:
```

—

3. Fragment 3

```
x = 3;
success=0;
while (!success) {                  Number of iterations: __________
   if (x) {x++;}
   if (x%2) {x++;}                  Output:
   if (x%3) {success++;}
   printf("Boo\n");
}
```

—

4. Fragment 4.

```
x = 1;
do {                  Number of iterations: __________
   printf("%d", x);
   x = x*2 - x;
   if (x) {x++;}       Output:
   else {x++;x++;}
}
while (x < 6);
```

—