16.216: ECE Application Programming

Practice Problems: Loops

1. What does each of the following programs print?

a.
```
int main() {
   int x = 0;
   while (x < 10) {
      printf("x = %d",  ++x);
      x++;
   }
   return 0;
}
```

b.
```
int main() {
   int i, j;

   for (i = 0; i < 3; i++) {
      printf("i is %d\n", i);
      for(j = 0; j < 5; j++)
         printf("i is %d, j is %d\n", i, j);
      printf("end of i = %d loop\n", i);
   }
   return 0;
}
```

c.
```
int main() {
   int x;
   int i = 0;

   for (x = 0; x <= 3; x++){
      printf("Start: x = %d, i = %d\n", x, i);
      x = x * 2;
      i++;
      printf("End: x = %d, i = %d\n", x, i);
   }
   printf("Final: x = %d, i = %d\n", x, i);
}
```

1 (cont.) What does each of the following programs print?

```
d. int main() {
     int x = 4;
     int n = 0;

     while (x > 5) {
       if (x == 10)
         x = 0;
       else
         x += 2;
       printf("x = %d\n", x);
       n++;
     }

     printf("n = %d\n", n);

     return 0;
   }
```

```
e. int main() {
     int x = 4;
     int n = 0;

     do {
       if (x == 10)
         x = 0;
       else
         x += 2;
       printf("x = %d\n", x);
       n++;
     } while (x > 5);

     printf("n = %d\n", n);

     return 0;
   }
```

```
f. int main() {
     int num = 625;

     while (num >= 1) {
       printf("num = %d\n", num);
       num /= 5;
     }
     return 0;
   }
```

2. Write a program to do each of the following tasks:
   (NOTE: You do not have to do any error checking in these programs unless the problem explicitly specifies that you do so.)

a. Print all multiples of 2 between 10 and 100, including the endpoints (i.e., print both 10 and 100).

b. Repeatedly prompt a user to enter two double-precision values, then read those numbers. Your program should end when the second number entered is less than the first—at that point, print `"Program complete"`. A sample run is below; user input is underlined:

```
Enter two values: 1 3
Enter two values: -0.7 1.234
Enter two values: 55 55
Enter two values: 16.216 16.217
Enter two values: 2.3 -3.7
```

c. Prompt for and read in a series of characters, stopping when the user enters the character `'q'`. Print the following outputs:
   - If the character is `'A'` or `'a'`, print `"Absolute value\n"`
   - If the character is `'C'` or `'c'`, print `"Cosine\n"`
   - If the character is `'S'` or `'s'`, print `"Sine\n"`
   - If the character is `'T'` or `'t'`, print `"Tangent\n"`
   - For all other characters, print `"Invalid input\n"`

d. Prompt for and read in a series of integers, and keep track of the largest and smallest values entered. Stop reading when the user enters a value outside the range 16 <= *n* <= 216; this final value should not be considered as the largest or smallest. After the user enters a value outside the range, print the largest and smallest values entered. A sample run is below:

```
Enter integer between 16 and 216: 17
Enter integer between 16 and 216: 216
Enter integer between 16 and 216: 53
Enter integer between 16 and 216: 1
Largest value: 216
Smallest value: 17
```

e. Prompt for and read in a series of characters and count the number of whitespace characters—spaces, tabs (`'\t'`) and newlines (`'\n'`)—in the list. Stop reading when the user enters the same non-space character twice in a row. Print the total number of whitespace characters. A sample run is below; it contains 1 tab, 3 spaces, and 2 newlines:

```
Enter input characters:
ab    3 6 ?
h Q
zz
Total whitespace characters: 6
```

(Note: tab is between 'b' and '3')

3. Write a program with a series of statements to produce each of the following patterns. Use only the following `printf()` statements. Each `printf()` may only appear once in each segment of the program. A sample template of the program appears on the next page.

```
printf("*");    printf(" ");    printf("\n");
```

```
a. ****
   ****
   ****
   ****
   ****
```

```
b. *
   ***
   *****
   *******
```

```
c. *****
   ****
   ***
   **
   *
```

```
d. *
   **
   ***
   ****
   *****
   ******
   *******
```

```
e. *********
   *********
   *********
   *********
   *********
   *********
   *********
```

```
f. *****
   *******
   *********
   ***********
   *************
```

```
g. *******
   ******
   *****
   ****
   ***
   ****
   *****
   ******
   *******
```

```
h. *************************
   ****************
   *********
   ****
   *
```

```
i. *************************
   **********************
   *****************
   **********
   *
   **********
   *****************
   **********************
   *************************
```

```
j. *
   ***
   *****
   *******
   *********
   ***********
   *********
   *******
   *****
   ***
   *
```

```
#include <stdio.h>
int main() {
   // declare variables as needed
   printf("----------------------------- Pattern 1\n");
   // code to produce pattern 1
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section
   printf("----------------------------- Pattern 2\n");
   // code to produce pattern 2
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section
   printf("----------------------------- Pattern 3\n");
   // code to produce pattern 3
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section
   printf("----------------------------- Pattern 4\n");
   // code to produce pattern 4
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section
   printf("----------------------------- Pattern 5\n");
   // code to produce pattern 5
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section
   // code and header for patterns 6, 7, 8
   printf("----------------------------- Pattern 9\n");
   // code to produce pattern 9
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section
   printf("----------------------------- Pattern 10\n");
   // code to produce pattern 10
   // printf(" "), printf("*"), and printf("\n")
   // may only appear once in this section

   return 0;
}
```