Study Questions: Set No. 5
Introduction to C
Saturday October 19, 2013

Covering:

Chapters 6 and 7

1. What output does the following program fragment produce?

```
i = 1;
while (i <= 128)
{ printf ("%d ", i );
  i *= 2;
}
```

2. What output does each of the following two statements produce?

```
for (i = 5, j = i - 1;
     i > 0;
     --i, j = i - 1)
        printf("%d %d", i, j);

for (i = 5, j = i - 1;
     i > 0, j > 0;
     --i, j = i - 1)
        printf("%d %d", i, j);
```

3. Which one of the following statements is not equivalent to the other two (assuming that the loop bodies are the same)?

- `while (i < 10) {…}`
- `for (; i < 10;) {…}`
- `do {…} while (i < 10) ;`

4. What output does the following `for` statement produce?

```
for (i = 10; i >= 1; i /= 2)
  printf("%d ", --i);
```

5. What output does the following `for` statement produce?

```
for (i = 10; i >= 1; i %= 2)
  printf("%d ", i++);
```

6. What output does the following `for` statement produce?

```
for (i = 10; i >= 1; i *= 2)
  printf("%d ", (i-=6));
```

7. What output does the following `for` statement produce?

```
for (i = 10; i >= 1; i -= 1)
  printf("%d ",(i/=2));
```

8. Predict the output of this program fragment:

```
i = 0;
while(i <= 5)
{
  printf("%3d %3d\n", i, 10 - i);
  i = i + 1;
}
```

9. What is displayed by this program fragment for an input of 8?

```
scanf("%d", &n);
ev = 0;

while (ev < n)
{
   printf("%3d", ev);
   ev = ev + 2 ;
}

printf("\n");
```

10. What output values are displayed by the following `while` loop for a data value of 5? Of 6? Of 7? In general, for a data value of any number *n*, what does this loop display?

```
printf("Enter an integer> ");
scanf("%d", &x);
product = x;
count = 0;

while (count < 4)
{
   printf("%d\n", product);
   product *= x;
   count += 1;
}
```

11. What values are displayed if the call to `printf` comes at the end of the loop instead of at the beginning?

12. The following segment needs revision. Insert braces where they are needed and correct the errors. The corrected code should take five integers and display their sum

```
count = 0;
while (count <= 5)
count += 1;
printf("Next number> ");
scanf("%d", &next_num);
next_num += sum;
printf("%d numbers were added; \n", count);
printf("their sum is %d.\n", sum);
```

13. Write a program segment that computes *1 + 2 + 3 + ... + (n - 1) + n*, where *n* is a data value. Follow the loop body with an `if` statement that compares this value to `(n * (n + 1)) / 2` and displays a message that indicates whether the values are the same or different. What message (to you think will he displayed?

14. Trace the execution of the loop that follows for `n = 8`. Show values of `odd` and `sum` after the update of the loop counter for each iteration.

```
sum = 0;
for (odd = 1;
     odd < n;
     odd += 2)
   sum = sum + odd;

printf("Sum of positive odd numbers less than %d is %d.\n",
       n, sum);
```

15. Consider the following definiations:

```
#define CBEGIN 10
#define CLIMIT -5
#define CSTEP   5
```

indicate what values of `celsius` would be at the end of the following `for` loops:

```
for(celsius = CLIMIT; celsius <= CBEGIN; celsius += CSTEP)
    ;

for(celsius = CLIMIT; celsius >= CBEGIN; celsius += CSTEP)
    ;

for(celsius = CSTEP; celsius >= CBEGIN; celsius += CLIMIT)
    ;

for(celsius = CLIMIT; celsius <= CSTEP; celsius += CBEGIN)
    ;
```

16. Rewrite the code shown in the previous question so the effect is equivalent but no increment/decrement operator appears in an expression with another arithmetic operator.

17. What errors do you see in the following fragment? Correct the code so it displays all multiples of `4` from `0` through `100`.

```
for mult4 = 0;
mult4 < 100;
mult4 += 4;
printf("%d\n", mult4);
```

18. Trace the following program fragment

```
j = 10;
for(i = 1; i <= 5; ++i)
{  printf("%d %d\n", i, j);
   j -= 2;
}
```

19. Rewrite the previous program fragment so that it produces the same output but uses `0` as the initial value of `i`.

20. Write a program to display a *centimeters-to-inches* conversion table. The smallest and largest number of centimeters in the table are input values. Your table should give conversions in *10-centimeter* intervals. One centimeter equals 0.3937 inch.

21. There are 9,870 people in a town whose population increases by 10 percent each year. Write a loop that displays the annual population and determines how many years it will take for the population to surpass 30,000.

22. What is displayed by the following program segments?

```
m = 3;
n = 5;
for(i = 1; i <= n; ++i)
{
  for(j = 0; j < i; ++j)
  { printf("*");
  }
  printf("\n");
}
```

23. What is displayed by the following program segments?

```
m = 3;
n = 5;
for(i = n; i > 0; --i)
{
   for(j = m; j > 0; --j)
   { printf("*");
   }
   printf("\n");
}
```

24. Show the output displayed by these nested loops:

```
for (i = 0; i < 3; ++i)
{
  printf("Outer %4d\n", i);
  for   (j = 0;    j < 2;     ++j)
  {  printf(" Inner%3d%3d\n", i, j);
  }
  for (k = 2; k > 0; --k)
  { printf(" Inner%3d%3d\n", i, k);
  }
}
```

25. Write a program that displays the multiplication table for numbers 0 to 9.

26. Rewrite the following code using a `do-while` statement:

```
sum = 0;
for (odd = 1; odd < n; odd = odd + 2)
sum = sum + odd;
```

27. Write nests of loops that cause the following output to be displayed. The size should be taken as input.

```
0 1 2 3 4 5
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

28. Write nests of loops that cause the following output to be displayed. The size should be taken as input.

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
0 1 2 3 4 5
0 1 2 3 4
0 1 2 3
0 1 2
0 1
0
```

29. Write nests of loops that cause the following output to be displayed. The size should be taken as input.

```
5
4 5
3 4 5
2 3 4 5
1 2 3 4 5
0 1 2 3 4 5
```

30. Write a program that uses `printf` to display the following picture on the screen.
The size should be taken as input.

```
    %         %
     %       %
      %     %
       %   %
        % %
         %
        % %
       %   %
      %     %
     %       %
    %         %
```

31. Write a program that displays the following pattern for any size (given by the user).

```
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
* * * * * * * *
 * * * * * * * *
```

Your program must use the following three `printf` statements only once each

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

No other `printf` statements are allowed.

32. Design an interactive input loop that scans pairs of integer numbers until it reaches a pair in which the first integer number evenly divides the second.

33. During execution of the following program segment, how many lines of asterisks are displayed?

```
for (i = 0; i < 10; ++i)
    for (j = 0;j < 5; ++j)
       printf("**********\n");
```

34. During execution of the following program segment:
(a) How many times does the first call to `printf` execute
(b). How many times does the second call to `printf` execute
(c) What is the last value displayed?

```
for(i = 0; i < 7; ++i)
{  for(j = 0;j < i; ++j)
     printf(" %4d", i * j);
   printf("\n");
}
```

35. What does the following code segment display? Try each of these inputs: 345, 82, 6. Then, describe the action of the code.

```
printf("\nEnter a positive integer> ");
scanf("%d", &num);
do
{  printf("%d ", num % 10);
   num /= 10;
} while (num > 0);
printf("\n");
```

36. Rewrite the program segment that follows, using a `for` loop:

```
count = 0;
i = 0;
while(i < n)
{  scanf("%d", &x);
   if(x == i)
      ++count;
   ++i;
}
```

37. Write a program that asks the user to enter a fraction, then reduces the fraction to lowest term:

```
Enter a fraction: 6/12
In lowest terms: 1/2
```

Your program should deal with cases like `-6/12`, `6/-12`, and `-6/-12`.
Hint: To reduce a fraction to its lowest term, first compute the greatest common divisor (GCD) of the numerator and denominator. Then divide both the numerator and denominator by the GCD.

38. Write a program that determined which of entered dates comes earlier on the calendar.
The user may enter any number of dates. The user will enter `0/0/0` to indicate that no more dates will be entered. If numbers of dates are less than two, the program should indicate that at least two dates are required.

```
Enter a date (mm/ dd/ yy):3/6/08
Enter a date (mm/ dd/ yy):5/17/07
Enter a date (mm/ dd/ yy):6/3/07
Enter a date (mm/ dd/ yy):0/0/0
5/17/07 is the earliest date
```

39. Write a program that accepts from the user *n* integer numbers, where *n* is entered by the user at the beginning of the program, and then print the largest and the smallest numbers. Do not store the input numbers in an array.

40. Write a program that accepts from the user *n* integer numbers, where *n* is entered by the user at the beginning of the program, and then print the largest and the second largest numbers. Do not store the input numbers in an array.

41. Which of the following are not legal constants in C?
Classify each legal constant as either integer or floating-point.

```
(a) 010E2
(b) 32.1E+5
(c) 0790
(e) 100_000
(f) 3.978e-2
```

42. Give the decimal value of each of the following constants.

```
(a) 077
(b) 0x77
(c) 0xABC
```

43. Which of the following are not legal types in C?

```
(a) short unsigned int
(b) short float
(c) long double
(d) unsigned long
```

44. If `c` is a variable of type `char`, which one of the following statements is illegal?

```
(a) i+= c; /* i has type int */
(b) c = 2 * c - 1;
(c) putchar(c);
(d) printf(c);
```

45. Suppose that `i` is a variable of type `int` , `f` is a variable of type `float`, and `d` is a variable of type `double`. What is the type of the expression `i * f / d`?

46. Suppose that `i` and `j` are variables of type `int` . What is the type of the expression `i / j + 'a'`?

47. Suppose that `i` is a variable of type `int`, `f` is a variable of type `float`, and `d` is a variable of type `double`. Explain what conversions take place during the execution of the following statement:

```
d = i + f;
```

48. Does the following statement always compute the fractional part of `f` correctly (assuming that `f` and `frac_part` are `float` variables)? If not, what's the problem?

```
frac_part = f - (int) f;
```

49. Evaluate the following expressions.

```
(a) (char) ('z' - 2)
(b) (char) (5 + 'D')
(c) 'D' - 'A'
(d) 'f' - 'A'
(e) '6' - '7'
```

50. Assume that a program contains the following declarations:

```
char   c = '\1';
short  s = 2;
int    i = -3;
long   m = 5;
float  f = 6.5f;
double d = 7.5;
```

Give the value and the type of each expression listed below.

```
(a) c * i
(b) s + m
(c) f / c
(d) c / d
(e) d / s
(f) f - d
(g) (int) f
```

51. Assume that a program contains the following declarations:

```
float x = 10.5, y = 7.2;
int m = 5, n = 2;
```

Give the value and the type of each expression listed below.

```
(a) x / (double) m
(b) x / m
(c) (double)(n * m)
(d) (double)(n / m) + y
(e) (double)(n / m)
```

52. What does this segment print?

```
for(ch = (int) 'd'; ch < (int) 'n'; ch += 3)
  printf("%c", (char) ch);
printf("\n");
```

53. Write a `for` loop that would print the alphabet in uppercase letters.

54. Write a program that translates an alphabetic phone number into numeric form:

```
Enter phone number: CALLATT
2255288
```

In case you don't have a telephone nearby, here are the letters on the keys: 2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PRS, 8=TUV, 9=WXY. If the original phone number contains nonalphabetic characters (digits or punctuation, for example), leave them unchanged:

```
Enter phone number: 1-800-COL-LECT
1-800-265-5328
```

You may assume that any letters entered by the user are upper case.

55. Write a program that counts the number of vowels (a, e, i, o, and u) in a sentence:

```
Enter a sentence: And that's the way it is .
Your sentence contains 6 vowels.
```

56. Write a program that calculates the average word length for a sentence:

```
Enter a sentence: It was deja vu all over again.
Average word length:3.4
```

For simplicity, your program should consider a punctuation mark to be part of the word to which t is attached. Display the average word length to one decimal place.

57. Write a program that computes the factorial of a positive integer:

```
Enter a positive integer:6
Factorial of 6:720
```

(a) Use a short variable to store the value of the factorial .What is the largest value of *n* for which the program correctly prints the factorial of n ?
(b) Repeat part (a), using an int variable instead.
(c) Repeat part (a), using a long variable instead.
(d) Repeat part (a), using a float variable instead.
(e) Repeat part (a), using a double variable instead.
(f) Repeat part (a), using a long double variable instead.
In cases (d)-(f), the program will display a close approximation of the factorial, not necessarily the exact value.