

CS 150 Final Review Topics and Questions

As always, you will be required to write program segments, functions, programs, or combinations of all three. Let's take a look at some example problems you could be expected to solve and their difficulty and NO the solution to these questions will not be posted but you are more than welcome to come to my office to discuss your solutions to any or all of these problems.

1. Look at the review questions given for exams 2 and 3.

2. Topics:

- Writing (or explaining) program segments dealing with:
 - i. if statements
 - ii. switch statements
 - iii. while loops
 - iv. for loops
 - v. do-while loops
- Expressions using relational and/or logical operators
- Advanced output setw, fixed, setprecision
- Rewriting if blocks as switch blocks and visa versa
- Rewriting loops in one form to another
- Nested loops (tracing or writing solutions to nested loop problems)
- Correcting syntax, logic, and/or runtime errors
- Writing or explaining data file segments including reading from and/or writing to files
- Tracing function calls with various parameter passing options (your favorite)
- Writing functions including passed by reference and passed by value parameters
- Functions calling other functions
- Make sure you understand the various data types and type casting
- Arrays (1D and 2D)
- Character processing
- Files: input and output files

3. (Easy) Given the following statements:

```
int x, y;  
cin >> x >> y;
```

- a. Write a single **if** statement that will print POSITIVE if both x and y are greater than zero and print NOT POSITIVE otherwise.
- b. Write a **for** loop that will print all numbers from y to x that are divisible by 4.
- c. Write a **while** loop to do the same thing.

4. (Easy) Ask the user for an even number that is greater than zero, and store it into y. Keep asking for input until the user gives an acceptable number.
5. (Medium) Declare a two dimensional array (10 rows, 10 columns) to store a multiplication table. The value stored at row A and column B in the array should be the solution to multiplying A by B.
 - a. Write a function to fill the array.
 - b. Write a function to display the table, nicely formatted, to the screen.
 - c. Write a function that will write the array to a file, 1 row per line, with each integer separated by a tab.
6. (Medium) The file **pi100.txt** contains the first 100 digits of Pi, one per line. You need to write a program to read from the file, digit by digit, and determine how many times each digit (0-9) appears in the file. You need to display a numeric count as well as a histogram as shown below. Display the output to the screen.

Output:

DIGIT	COUNT	

0	8	*****
1	8	*****
2	12	*****
3	12	*****
4	10	*****
5	8	*****
6	9	*****
7	8	*****
8	12	*****
9	14	*****

7. (Medium) Write a program that will simulate the roll of a dice 6000 times and show the frequency in which each side appeared. Example output is below:

Face	Frequency
1	1003
2	1017
3	983
4	994
5	1004
6	999

8. (Easy): Write a function **computeEven** that accepts an array **values** and the number of values currently in the array, **howmany**. The function **computeEven** returns through the function name the number of values in the array that are even. Your function might be called by a statement of the form:

```
cout << "Number of evens = " << computeEven (values, howmany) << endl;
```

9. (Easy): An integer array **values** has been defined and filled beginning at subscript 0 with **howmany** elements in the array.

a) Write a prototype for a void function **insert** that will accept the array **values**, **howmany** elements in the array, and a new integer value to be inserted at the end of the array, **value**.

b) Write the actual function definition **insert** described in a) completely.

10. (Hard): Assume the array in P#9 is sorted from smallest to largest. Modify the function **insert** (above) to insert the value into the array in the proper position of the array preserving the sorted nature of the array.

11. (Medium): Write a function **numEvens** that accepts the array from P#9 and returns the number of even integers in the array. Use the function **computeEven** from problem #8.

12. (Medium): What will the following C++ program print?

```
#include <iostream>
void change (int&, int, int&);
void main()
{
    int i, j, k;
    i = 22;
    j = 4;
    k = 7;
    change (i, j, k);
    cout << i << " " << j << " " << k << " " << endl;
    change (k, i, j);
    cout << i << " " << j << " " << k << " " << endl;
}
void change (int& i, int j, int& k)
{
    i = i + k;
    j = i / 3;
    k = i % j;
}
```

13. (Hard): Many infinite series have sums equal to pi. One formula goes as follows:
 $\pi = 4 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + \dots$

This sequence consists of alternating positive and negative terms. You are to write a single C++ function which estimates pi using the above formula. The function might be called by a statement of the following form:

```
cout << "Pi = " << computePi(numterms) << endl;
```

where numterms is an integer value containing the number of terms to be used from the above sequence.

14. (Easy): Given the following prototypes and declarations, place a check by only the illegal function calls in the following list AND specify what caused the error.

```
int compute (int, int&, int);  
float doit (float&, int, float, char&);  
  
void main()  
{  
    int i, j;  
    int nums[10];  
    float x, y;  
    char ch;
```

_____ a) compute (i, j, i + j);

_____ b) compute (i, nums[1], char ch);

_____ c) doit (x, i, y, 'c');

_____ d) doit (x + y, nums[i], y, ch);

_____ e) doit (x, 3, x, static_cast<char>(static_cast<int>('a') + 1));

15. (Medium): Write a program segment which inputs a character value into the variable **ch** from the keyboard in the range of '0' to '9' inclusive. With one assignment statement, convert the value in **ch** to the numerical value of the digit and place this value in the integer variable val. As an example, if ch = '0', then val would equal the integer 0.

16. (Medium): Will the following program segment count and print the number of lines in a data file? Why or why not? If not, what does the program do?

```
int count = 0;
char ch;

inputData.get (ch);
while (!inputData.eof())
{
    if (ch == '\n')
    {
        count++;
    }
    else
    {
        inputData.get (ch);
    }
}
cout << count << endl;
```

17. (Easy): Which of the following declarations are legal?

- a) `int arry1[5] = {5, 4, 3, 2, 1, 0};`
- b) `int arry2[5] = {1, 2};`
- c) `char ch[5] = {'12345'};`

18. (Hard): An array **values** exists containing **howmany** elements. Write a function **anyDuplicates** that returns true if there are any duplicate values in the array; otherwise, return the value false. A call to your function might be as follows:

```
bDuplicates = anyDuplicates (values, howmany);
```

19. (Easy): Create a 2D array **values** with 4 rows and 3 columns. Place values of your choosing into the array.

20. (Medium): Write a function **writeValues** that accepts the array created in 19. along with the number of rows and columns. You are to write each value from the passed in array to a file `data.txt` one row at a time with each value separated by a single space.

21. (Hard): Rework problem 18. but instead of passing in a 1D array, you are to accept the 2D array from 19. and the number of rows and columns.

- a) Write the function prototype.

b) Write the function definition.

c) Show what a call to your function might look like.

22. (Medium): Write a program segment that prints to the screen the number of characters (including whitespace) in the data file story.txt.