While and Do While Loops

1. While Loop

The keyword while is used to perform one or more statements many times and the number of times is not known beforehand. The test occurs at the **beginning** of the loop.

```
int main () {
    int n=3;

    while (n != 0) {
      cout << "Enter a number (0 to end)? ";
      cin >> n;
    }
    cout << endl << "the END" << endl;

    return 0;
}
```

And here is the output:

```
Enter a number (0 to end)? 4
Enter a number (0 to end)? 3
Enter a number (0 to end)? 9
Enter a number (0 to end)? 0

the END
```

2. Do While Loop

The keyword do while is used to perform one or more statements many times and the number of times is not known beforehand. The test occurs at the **end** of the loop.

```
int main () {
    int n;

    do{
      cout << "Enter a number (0 to end)? ";
      cin >> n;
    } while(n != 0);
    cout << endl << "the END" << endl;

    return 0;
}
```

3. *Exercises* (Problems with an asterisk are more difficult)

1. Use the while loop command to do exactly the same thing as the following program segment. In other words, replace the for loop with the while loop.

```
for (int i=0; i<10; i++) {
     cout << i endl;
}
```

2. Write a program to keep asking the user to enter a number until the number entered in is a zero. The program terminates when the user enters a zero.

3. Just like question 2 but in addition when the program terminates, it will print out how many numbers the user has entered in.

4. Just like question 2 but in addition when the program terminates, it will print out the sum of all the numbers the user has entered in.

5. Just like question 2 but in addition when the program terminates, it will print out the average of all the numbers the user has entered in.

6. Just like question 2 but in addition when the program terminates, it will print out how many even numbers and how many odd numbers the user has entered in.

7. Write a program to enter two numbers, *num1* and *num2*. *num1* is divided by *num2* and the result is displayed. But before the division, *num2* is tested for the value 0. If it is a 0, the division does not take place and the program will keep asking the user to enter another number until a non-zero number is entered for *num2*.

8. Write a program to generate two random numbers between 0 and 100. Ask the user to enter the answer for the sum of the two numbers. Print out whether the answer entered by the user is correct or not. See the Random Number document on how to generate random numbers. If the answer is wrong then keep repeating to ask the user to enter the correct sum.

9. Write a program to determine how much a person would get if s/he is given one penny the first day, two pennies the second day, four pennies the third day, and continues to double each day until day 30. Display a table showing the day, how much for each day and the accumulated sum up to that day for the 30 days. Finally show the total amount in dollars at the end that the person would get.

10. Would you prefer to get $10,000 a day for 30 days or the payout as described in question 9? Write a program to determine which is the better option.

11. The Number Guessing Game – version 1 Player guesses. Write a program for the computer to pick a random number between 0 and 99. The program will then repeatedly ask the user to guess the number. Keep repeating until the guess is correct.

Note: the next two questions (12 and 13) are hard and very hard. Please do not feel discouraged if you can't get it, and definitely DO NOT search the web for the answer or ask your friend for help. It is more important and much better for you to spend two hours on it and still can't get it then to use two minutes to find the answer on the web to hand in to me.

12. ** The Number Guessing Game – version 2 Computer guesses. First, you as the user, pick a number between 0 and 99. The computer will then repeatedly make a guess by printing it out. After each guess, the user will enter whether the guess is correct or not. Keep repeating until the guess is correct. At the end, print out how many times the computer needs to guess the number.

13. **** The High-Low Number Guessing Game – version 3 Computer guesses. Similar to question 12, but instead of the user just telling the computer whether the guess is correct or not, the user can also tell the computer whether its guess is too high (i.e. the guess is higher than the number) or too low (i.e. the guess is lower than the number). Based on this, the computer should be able to guess the number in fewer tries then for the solution for question 12.

