MTH 3300
E. Fink

3300 Problems, Section 5: Loops

1. Write code that will print out the result of the sum $1^2 + 2^2 + 3^2 + 4^2 + \ldots + 100^2$. (4 lines)
2. Write code that will print out the result of the sum $1 + 4 + 7 + 10 + \ldots + 97 + 100$. (4 lines)
3. Create a `while` loop which will output the first 50 EVEN integers, each on a different line. (4 lines)
4. Write a loop that prints out all multiples of 7 less than 1000. (3 lines)
5. Suppose that the user has *already* entered a positive integer `n`. Write the code necessary to print out the product of all the EVEN integers LESS than `n`. (For example, if the user had entered `7`, the program should print out `48`, because $2 \times 4 \times 6 = 48$.) (4 lines)
6. Suppose that the user has *already* entered a positive integer `n`. Write the code necessary to print out the sum of all the ODD integers LESS than `n`. (For example, if the user had entered `8`, the program should print out `16`, because $1 + 3 + 5 + 7 = 16$.) (4 lines)
7. Determine what the code below displays:

```
int i=1, j=20;
while(i<j)
{
    i+=2;
    j--;
}
cout << j;
```

8. Explain in words what interesting phenomenon happens when the following loop is run (and why), in one sentence.

```
int x = 1;
while(x <= 10)
{
      cout << x << endl;
}
```

```
int x = 2, y = 3;
while ( y < 10 )
{
      x++;
      if ( x != y )
      {
            cout << x;
      }
      y += 3;
      cout << y << endl;
}
```

9. What will print out from the following loop?

```
int x = 2, y = 3;
while ( y < 10 )
{
      x++;
      if ( x == y )
      {
            cout << x;
      }
      y += 3;
      cout << y << endl;
}
```

10. Determine what the code below displays:

```
int i=1, j=2;
while(i<20)
{
      i+=j;
      j++;
      if(j==4)
      {
            break;
      }
}
cout << i;
```

11. The following loop contains no *syntax* errors, but there is still an issue with the way that it is coded. Briefly explain what will happen when you run this code:

```
int i = 1;
double x = 0;
while(i!=10)
{
    x += i;
    i += 2;
}
cout << x << endl;
```

12. Determine what the code below displays:

```
int i=1, j=1;
while(i<10)
{
    i+=j;
    j++;
}
cout << j;
```

13. Determine what the code below displays:

```
int x = 1, y = 1;
while(x + y < 5)
{
      x = x - 1;
      cout << y << endl;
      y = x + y + 3;
      cout << x;
}
```

14. Using a `for` loop, write code that will print out the sum of the first 100 positive odd numbers. (4 lines)

15. Using a `for` loop, write C++ statements which can calculate the sum of the first 20 terms of the series $1 + 2 + 4 + 8 + 16 + \ldots$. You may assume that any useful libraries have been included. (3 lines)

16. Using a loop, write code that will print out 100 lines that look like this (2 lines):

```
1.  Hello
2.  Hello
3.  Hello
```

. . . and so on, until it finally prints out

```
100.  Hello
```

17. Write code that will print out the numbers between 1 and 100, except that: instead of the number, it will print `Bop` if the number is a multiple of 2, `Slap` if the number is a multiple of 3, and `BopSlap` if the number is both. So, the printout should start like:

```
1

Bop

Slap

Bop
```

```
5

BopSlap

7
```

(9 lines)

18. Write the code necessary to do the following. Print out the first 1000 numbers, one on each line, EXCEPT: instead of printing the number, print “Fizz” if the number is a multiple of 3, “Buzz” if the number is a multiple of 5, and “FizzBuzz” if the number is a multiple of both 3 and 5. (8 lines)

19. *Describe* what the code below displays (careful!):

```
int x = 1, y = 1;
for(int i = 1; i<10; i++)
{
    x = x + i;
    cout << x;
    i = y;
}
```

20. The following code is run to sum the first 10 integers. The displayed value ends up being `-104015015`. Why?

```
int main()
{
      int sum;
      for(int i=1; i<=10; i++)
      {
            sum += i;
      }
      cout << sum << endl;
      return 0;
}
```

21. Consider the code snippet below.

```
ifstream file;
file.open("numbers.txt");
bool cool = true;
int x;
while(cool && (file >> x))
{
      cool = !cool || (x > 2);
      cout << x;
}
```

If the file `numbers.txt` contains `3   5   2   8`, what will be printed to the console?

22. What would the following code display?

```
bool flag = false;
int i = 10;
while(!flag)
{
    i = i+5;
    if(i >25)
    {
        cout << i << endl;
        flag = true;
    }
    cout << i << endl;
}
```

23. What will print out from the following loop?

24. Consider the code snippet below.

```
bool donezo = false;
while(!donezo)
{
      int x;
      cin >> x;
      if( x == 2 )
      {
            donezo = (donezo || (x == 2));
      }
      cout << x;
}
```

If the user enters `3   5   2   8` while this portion of the code is running, describe what will be printed to the console.

25. The following loop uses a flag to keep asking the user to input an integer, until they input an integer that is **between 1 and 100**. What lines should fill in the blank so that the flag works properly? *Assume that the user always inputs an integer.* (2 lines)

```
bool success = false;
int entry;
while( !success )
{
      cout << "Enter an integer (but if it’s not between 1 and 100, I’ll just ask again):  ";
      cin >> entry;
      ____________________
      ____________________
      ____________________
      ____________________
}
```

26. What value will print out when the following code runs?

```
int x = 0;
for(int i = 1; i <= 2; i++)
{
      for (int j = 1; j <= 2; j++)
      {
            x = x + j;
      }
      cout << i << endl;
}
cout << x;
```

27. What will print out when the following code runs?

```
for(int i = 1; i <= 3; i++)
{
      for (int j = i+1; j <= 3; j++)
      {
            cout << i+j;
      }
      cout << endl;
}
```

28. What does the following code print out?

```
for(int r = 1; r<=4; r++)
{
       for(int c = 1; c<=r; c++)
       {
             cout << "X";
       }
       cout << endl;
}
```

29. What will print out from the following?

```
int x = 1;
while(x<5)
{
      for(int i = 1; i <= x; i++)
      {
            cout << i;
            x+=i;
            break;
      }
      cout << ",";
}
```

30. What will the following nested for loops display?

```
for(int i=1; i<=3; i++)
{
    for(int j=1; j<=4; j++)
    {
        cout << i+j << " ";
    }
}
```

31. What is printed out by the following code? Be precise!

```
for(int k = 1; k<=3; k++)
{
      for(int j = 1; j<=k; j++)
      {
            cout << k + j << endl;
      }
      cout << "Hey" << endl;
}
```

32. What value will print out when the following code runs?

```
int x = 0;
while(x < 4)
{
      for (int i = 1; i <= 3; i++)
      {
            x += i;
            cout << i << " ";
      }
      cout << x << endl;
}
cout << x;
```

Longer Problems

1. Write the entire `main()` function of the following program. (Around 8 lines)

   The program should open a file named `mobydick.txt`. The program should count the number of times the letter “e” appears in the file, as either a lowercase or uppercase letter.

   For example: if `mobydick.txt` contains

   ```
   Evan excels
   ```

   then `3` should print to the console.

2. Write the entire `main()` function of the following program. (Around 15 lines)

   The program should open a file named `numbers.txt`. This file will contain a list of different numbers. The program should read the file, and then write all the positive numbers to a file called `positive.txt`, and write all the negative numbers to a file called `negative.txt`.

   For example: if `numbers.txt` contains

   ```
   -23.4
   2
   0
   3.1
   -15
   ```

   then `positive.txt` should contain

   ```
   2
   3.1
   ```

   afterwards, while `negative.txt` should contain

   ```
   -23.4
   -15
   ```

3. Write the entire `main()` function of the following program. (Around 8 lines)

   The program should ask the enter 10 *positive* integers, and then print out the *largest one that is LESS THAN 100* to the console. Use a LOOP for full credit. You may assume that all numbers are positive, and at least one number is less than 100.

   Example run (where the numbers in the first row are user input):

   ```
   Enter five integers:  30 5 140 98 100 10 55 120 40 96
   The largest less than 100 is:  98
   ```

4. Write the entire `main()` function of the following program. (12 lines)

   Write a program that first asks the user to enter a “special” letter. The program should then ask the user to enter a single line sentence containing all lowercase letters and no punctuation, except for a **period at the end**. The program will then output the number of times that this letter appears in the sentence.

   Example:

   ```
   Enter a special letter:  t

   Enter a sentence:  here is my little sentence.

   Your letter appears 3 times.
   ```

   (`t` and `here is my little sentence.` are user inputs, the rest is output by the program.) **HINTS**: read the “special” letter in first. Then read the sentence in character by character, instead of to a `string`. Also, a `do ...while` loop may be useful, but certainly is not necessary – although some type of loop will be.

5. Write the entire `main()` function of the following program. (9 lines)

   Write a program that first asks the user to enter a single line sentence containing all lowercase letters and no punctuation, except for a **period at the end**. The program will then output the number of letters in the sentence, not including spaces or the period. (You do not have to explicitly exclude the spaces – if you do this in any way like we’ve done in class, the spaces will be automatically ignored.)

   Example:

   ```
   Enter a sentence:  hey there.
   ```

```
Your sentence has 8 letters.
```

(`hey there.` is user inputs, the rest is output by the program.) **HINTS**: Read the sentence in character by character, instead of to a `string`. Also, a `do ...while` loop may be useful, but certainly is not necessary – although some type of loop will be.

6. Write the entire `main()` function of the following program. (Around 13 lines)

   Simulate a 2D drunkard's walk with 100 steps. That is, write a program that starts at $(0, 0)$, and at each step, goes either $+1$ in the x-direction, -1 in the x-direction, $+1$ in the y-direction, or -1 in the y-direction, each possibility happening with probability 1/4.

   For example, the first few steps might be $(0, 0) \to (0, 1) \to (-1, 1) \to (-2, 1) \to (-2, 0) \to (-1, 0) \to (-2, 0)$

   Have the program print out the location (that is, the x-coordinate and y-coordinate) after 100 steps.

7. Write the entire `main()` function of the following program. (Around 7 lines)

   The program asks for a word (all lowercase, with no spaces), and then counts the number of *character doubles* – that is, two of the same character in a row. For example, if the user enters `bookkeeper`, the program would output 3, because of "oo", "kk", and "ee". If the user enter `butterball`, the program would output 2, because of "tt" and "ll". You may assume that the user will never enter a word with three or more of the same characters in a row.