COMP 1602: Computer Programming II

Lab #4: Characters and ASCII

1. Suppose that *c* is a character variable. If the character '8' was assigned to *c*, what value is assigned to *d* by the following statement?

```
int d;
d = c – '0';
```

2. What is the output produced by the following fragments of code, if *c* is a character variable?

a.
```
c = 87;
cout << "c = 87 is: " << c << endl;
```

b.
```
c = 87;
c = c + 10;
cout << "(c = 87) + 10 is: " << c << endl;
```

c.
```
c = 'm' - 10;
cout << "'m' - 10 is: " << c << endl;
```

d.
```
c = '1' * 2;
cout << "'1' * 2 is: " << c << endl;
```

e.
```
c = '5' - 5;
cout << "'5' - 5 is: " << c << endl;
```

f.
```
c = '\0';
c = c + 64;
cout << "(c = '\\0') + 64 is: " << c << endl;
```

3. a. Write a function *isVowel* which accepts a character *c* as a parameter and returns *true* if *c* is a vowel and *false* otherwise. The function should work for both uppercase and lowercase letters.

b. Using the *isVowel* function, write a program which reads a character entered by the user at the keyboard and prints "*Is a vowel*" if the character is a vowel and "*Is not a vowel*" if it is not.

4. Write a program which reads a passage from a file, *input.txt*, and displays the number of characters which are present in each of the following categories:

- Digits
- Lowercase letters
- Uppercase letters
- Characters which are neither digits nor letters

Your program should use functions to check for digits, uppercase letters, and lowercase letters.

5. Write a program to generate (i.e., not hard code) an alphabet chart as follows:

```
Z z   Y y   X x   W w   V v
U u   T t   S s   R r   Q q
P p   O o   N n   M m   L l
K k   J j   I i   H h   G g
F f   E e   D d   C c   B b
A a
```

6. Write a function, *charToBinary*, which given a character, *c*, and an array of 8 integers, *byte*, as parameters, converts the ASCII equivalent of *c* to its binary representation and stores it in *byte*. For example, if *c* is 'a', *byte* should be 01100001. The heading of the function is as follows:

```
void charToBinary (char c, int byte[])
```

7. Write a function, *binaryToChar*, which given an array of 8 integers, *byte*, containing the binary representation of an ASCII character as a parameter, returns the ASCII character. For example, if *byte* is 01100001, the function should return 'a'. The heading of the function is as follows:

```
char binaryToChar (int byte[])
```

8. Write a function, *getPosition*, which accepts a character as a parameter and if the character is a letter, returns the position of the letter in the alphabet (as a number between 1 and 26, inclusive); the function should return 0 if the character is not a letter. For example, *getPosition* ('#') should return 0, *getPosition*('A') should return 1, *getPosition*('a') should return 1, and *getPosition*('Y') should return 25, etc.

NB: The position of an uppercase letter can be found by subtracting 'A' and adding 1. The position of a lowercase letter can be found by subtracting 'a' and adding 1.