CMPT 125 Assignment 2

- Due Jan. 23 at 3:20pm; please hand in hard copy in assignment box in CSIL
- You may write or type your answers

Question 1

(20 marks total)

a) Let's consider an integer array of size 10. (10 marks, each part is 2 marks)

```
int a[10];
```

I. How would you assign a pointer, called `pA`, to store the address of element 0 of this array? Write the C code for your answer.
II. Using `pA`, how would you obtain the value of the next element (element 1) of the array?
III. Explain in 2-3 sentences what the statement `pA = a[1];` would do.
IV. Is the statement `a = pA;` valid (would it cause compilation errors)? How about `a++;`? Explain briefly in 3-4 sentences.
V. Write a C code to print all the elements from this array.

b) Consider the two variables below. (5 marks)

```
char amessage[] = "now is the time";

char *pmessage = "now is the time";
```

I. What is the difference between the two variables `amessage` and `pmessage`? (1 mark)
II. Write a C code to change the character `'t'` from `'time'` to uppercase `'T'` in the variable `amessage`. (5 marks)

c) Consider the following code. (5 marks)

```
void strcopy(char s1[], char s2[])
{
  int len = strlen(s2);

  for (int i = 0; i<=len; i++) {
    s1[i] = s2[i];
  }
}

int main () {
  char s2[10] = "copy this";
  char s1[10];

  strcopy(s1, s2);
}
```

The above code copies the character from `s2` to `s1` using the concept of array, in the `strcopy` function. Implement another function `strcopy2` that achieves the same result, but that takes as input pointers to character arrays – that is, complete the `strcopy2` function in the following code so that the code copies the content of `s2` into `s1`.

```
void strcopy2(char *s1, char *s2) {
}

int main () {
  char s2[10] = "copy this";
  char s1[10];

  strcopy2(s1, s2);
}
```

Question 2

(10 marks, each part is 2 marks)

Assume that each of the expressions below gives the processing time $T(n)$ spent by an algorithm for solving a problem of size n. Find the dominant term(s) having the steepest increase in n and specify the lowest Big-O complexity of each algorithm.

Expression	Dominant term(s)	O(…)
$0.001 \log_4 n + \log_2(\log_2 n)$		
$n^3 \log_2 n + n(\log_2 n)^2$		
$0.01n + 100n^3$		
$0.1n + 4n^{1.5} + 2.5n^{1.5}$		
$3 \log_8 n + \log_2(\log_2(\log_2 n))$		

Question 3

(8 marks, each part is 2 marks)

For the following parts, try to get the best Big-O estimate that you can and briefly justify your answers.

Part a)

```
int i, j;
int n = 100;

for (i = 1; i <= n; i++) {
   for (j = 3*i; j <= n; j++) {
       printf("programming is fun\n");
   }
}
```

Part b)

```
int i, j;
int n = 1000000;

for (i = 1; i <= n; i++) {
   for (j = 1; j <= 10000; j++) {
       printf("%d %d\n", i, j);
   }
}
```

Part c)

```
int i = 0;
int n = 10;
int j;

while (i < n) {
   i++;
   j = i;
   while (i < n) {
       printf("hello %d\n", i);
       i++;
   }
   i = j;
}
```

Part d)

```
int i = 0;
int n = 10;
int j;
```

```
while (i < n) {
   i++;
   j = i;
   while (i < n) {
       printf("hello %d\n", i);
       i++;
       break;
   }
   i = j;
}
```

Question 4

(15 marks)

Write a program in C that prompts a user to select any five beverages of your choice (e.g Coke, Lemon Tea, etc.). User should be able to select one of these options using only **integers from 0 to 4** as inputs. Once a user has made the choice, print the chosen drink along with its calorie information. You may look up the calorie information online.

Example input: 3

Example output:

```
Selected drink: Coca Cola
Calorie count: 140 Calories
```

Notes:

- You can assume that the user will input integers. You do not need to handle invalid inputs of other types (such as strings, floats, etc.).
- Your program should gracefully terminate if a user enters wrong input three times. In addition, after every invalid attempt, your program should tell the user how many tries he/she has remaining. For example, the prompt should work something like this:
  - Prompt 1: User enters 6. Your program should tell that input is invalid and enter again. Also print the remaining tries (2).
  - Prompt 2: User enters -1. Your program should tell that input is invalid and enter again. Also print the remaining tries (1).
  - Prompt 3: User enters 200. Your program should tell that input is invalid and terminate. Also print the remaining tries (0).
  - If the user enters a correct input on or before the last try, the program should behave normally, and output should be like the provided example.