COMP2510 – Review Exercise 1 (Winter 2017)

1. Implement the following functions without calling any other function:

   (a) a function

   ```
   size_t find_first(const char s[], int c);
   ```

   that returns the index of the first occurrence of the character `c` in the string `s`. If `c` is not found, `find` returns -1 (cast as `size_t`).

   (b) a function

   ```
   void replace_first(int a[], size_t n, int oldint, int newint);
   ```

   that replaces the first occurrence of the integer `oldint` in the array `a` of `n` integers by the integer `newint`.

   (c) a function

   ```
   void replace_last(char s[], int oldc, int newc);
   ```

   that replaces the last occurrence of the character `oldc` in the string `s` by the character `newc`.

   (d) a function

   ```
   size_t count(const int a[], size_t n, int x);
   ```

   that returns the number of occurrences of the integer `x` in the array `a` of `n` integers.

2. Implement the following functions. You may call functions in the standard C library in your implementation.

   (a) a function

   ```
   size_t count_alpha(const char s[]);
   ```

   that returns the number of alphabets in the string `s`.

   (b) a function

   ```
   int all_digits(const char s[]);
   ```

   that returns 1 if the string `s` consists entirely of digits; otherwise, it returns 0.

   (c) a function

   ```
   void lowercase_copy(char dest[], const char src[]);
   ```

   that copies the string `src` to `dest` with all uppercase characters changed to lowercase in the copy. Assume that `dest` is large enough to store the copy.

   (d) a function

   ```
   void reverse_copy(char dest[], const char src[]);
   ```

   that copies the string `src` to `dest` in reverse. Assume that `dest` is large enough to store the copy. For example, after executing the following

   ```
   char s[10]; reverse_copy(s, "hello");
   ```

   `s` will contain the string `"olleh"`.

   (e) a function

   ```
   void ltrim_copy(char dest[], const char src[]);
   ```

   that makes a copy of `src` with leading whitespaces removed & store the copy in `dest`. (A whitespace is any character that tests true by the `isspace` function.) Assume that `dest` is large enough to store the copy.

(f) a function

```
int is_valid_id(const char s[]);
```

that tests whether the string s is a valid BCIT student ID. For simplicity, a valid student ID is a string that starts with either an ‘a’ or ‘A’ & is then followed by exactly 8 digits. The function returns 1 if s is a valid student ID; otherwise, it returns 0.

3. Write a C program that counts the number of alphabets in a text file. The name of the text file is specified on the command-line. The program should check that exactly 1 file is specified.

4. In each of the following, indicate whether the statement marked (@) is valid. (We regard a statement as invalid if either it won’t compile or if it may cause a runtime error.) If the statement is invalid, explain why; if the statement is valid, indicate the output of the printf statement that follows.

(a)

```
char a[]  = "hello";
char *p   = "world";
p = &a[1]; /* (@) */
printf("%s", p);
```

(b)

```
char a[]  = "hello";
char *p   = "world";
p[1]  = *a; /* (@) */
printf("%s", p);
```

(c)

```
char a[]  = "hello";
char *p   = "world";
a = p; /* (@) */
printf("%s", p);
```