Lesson 9 Lab

Student Name: Markus Afonso **Student ID: A01333486**

Task 1: To define a program which reads in a word and then to print it in all capital letters. Example:

input: apple -> output: print APPLE
input: intel -> output: print INTEL

[Hint: to check out ASCII table to find out the distance between ‘a’ and ‘A’, this distance applied to all letters, between ‘b’ and ‘B’, between ‘c’ and ‘C’ and so on]

Requirement: you are not allowed to use any function defined in <ctype.h>

```
#include <stdio.h>

int main()
{
    char input[100];
    scanf("%s", input);

    for (int x = 0; input[x] != '\0'; x++ )
    {
        if (('a' <= input[x]) && ('z' >= input[x]))
        {
            input[x] = input[x] - ('a' - 'A');
        }
    }
    printf("%s", input);

    return 0;
}
```

Task 2: To define a program reads in a string and then prints “yes” or “no” based on whether it’s a valid BCIT ID, let’s use standard-library

[Hint: valid BCIT IDs need to be 9 chars long with first char as ‘a’ or ‘A’ followed by 8 digits]

Examples:

A00123456 -> yes
a12345678 -> yes
aa1234567 -> no (why: 2nd char should be digit)
A123456789 -> no (why: too long – it has 10 chars in total)

```
#include <stdio.h>
#include <ctype.h>

int main()
{
    char input[100];
    scanf("%s", input);
    int x = 0;
    for (; input[x] != '\0'; x++)
    {
        if (!isdigit(input[x]) && x != 0)
        {
            printf("no");
            return 0;
        }
    };
    if (x != 9)
    {
        printf("no");
        return 0;
    }

    if ((input[0] != 'a') && (input[0] != 'A'))
    {
        printf("no");
        return 0;
    }
    printf("yes");
    return 0;
}
```

Task 3: (Same question as Part 1) To define a program which reads in a word and then to print it in all capital letters, however, this time you are required to use standard-library

```
#include <stdio.h>
#include <ctype.h>
```

```
int main()
{
    char input[100];
    scanf("%s", input);

    for (int x = 0; input[x] != '\0'; x++ )
    {
        if (islower(input[x]))
        {
            input[x] = toupper(input[x]);
        }
    }
    printf("%s", input);

    return 0;
}
```

Task 4: To define a program which hard-codes the given data as below (not to read from user), use the standard library to converts them to numbers, and then prints their sum.

Given data:

char num_in_chars_1[] = "123";
char num_in_chars_2[] = "4567890";

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char num_in_chars_1[] = "123";
    char num_in_chars_2[] = "4567890";

    int num1 = atoi(num_in_chars_1);
    int num2 = atoi(num_in_chars_2);
    printf("%d", num1 + num2);

    return 0;
}
```

Task 5: To define a program which reads in a string from user, and then print it with each char three times.
Example:

input: “hello!”
output: “hhheeellllllooo!!!”

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char input[100];

    fgets(input, 100, stdin);

    for (int x = 0; input[x] != '\0'; x++)
    {
        printf("%c%c%c", input[x], input[x], input[x]);
    }

    return 0;
}
```

Task 6: To define a program which reads from a hardcoded string “123 456 789”, (NO user input) using function sscanf, and then using function sprintf to print ”the result is %d” (the sum of the three numbers) into a string (named s), at the last print out s.
[Hint: Fairly similar to the combination of the two examples of sprintf and sscanf]

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char string[] = "123 456 789";
    int a;
    int b;
    int c;
    char s[100];
    sscanf(string, "%d%d%d", &a, &b, &c);
    sprintf(s, "The result is %d", a + b + c);
    printf("%s", s);
    return 0;
}
```