COMP 2510

Lab 6

You are expected to copy and paste your code into each corresponding box in this handout and submit it as a Word document or PDF file before the due. Additionally, your lab instructor will tell you which three questions you must showcase during the lab session. While you may demonstrate your code running in person after the due date, your file must be submitted on time.

Hint #1: For questions asking for 'a **function** which takes in xyz', the xyz refer to function parameters, not the use of scanf().

Hint #2: For questions asking for 'a **program** which reads in xyz', the xyz refer to the use of scanf().

Hint #3: For questions asking for 'a program', you are expected to include all necessary #include and define your main function. It's up to you whether to define additional helper functions, though usually, you don't need to.

Task 1 Part 1: Write 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(void) {
    char input[128];
    printf("Enter a string (max 128 string):\n");
    scanf("%s", input);

    for (size_t i = 0; input[i] != '\0'; ++i) {
        char curr = input[i];
        printf("%c",
                curr >= 'a' && curr <= 'z'
                ? curr + ('A' - 'a') : curr
            );
    }
    printf("\n");

    return 0;
}
```

Task 1 Part 2: Same question as Part 1, however, this time you are required to use standard-library

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

int main(void) {
     char input[128];
     printf("Enter a string (max 128 string):\n");
     scanf("%s", input);

     for (size_t i = 0; input[i] != '\0'; ++i) {
     char curr = input[i];
     printf("%c", islower(curr) ? toupper(curr) : curr);
     }
     printf("\n");

     return 0;
}
```

Task 2: Write a **program** which hardcodes 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(void) {
	char num_in_chars_1[] = "123";
	char num_in_chars_2[] = "4567890";

	printf("%i\n", atoi(num_in_chars_1) + atoi(num_in_chars_2));

	return 0;
}
```

Task 3: Write a **program** reads in a string and then prints “yes” or “no” based on whether the input is 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)

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

int main(void) {
     char input[32];
     printf("Enter a string (max 32 characters):\n");
     scanf("%s", input);

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

     size_t i = 1;
     while (input[i] != '\0') {
     char curr = input[i];
     if (!isdigit(curr)) {
          printf("no\n");
          return 0;
     }
     ++i;
     }

     printf("%s\n", i == 9 ? "yes" : "no");
     return 0;
}
```

Task 4: Write a **program** which reads in a string from user, and then print it with each char three times.
Example:

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

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

int main(void) {
     char input[32];
     printf("Enter a string (max 32 characters):\n");
     scanf("%s", input);

     size_t i = 0;
     while (input[i] != '\0') {
         char curr = input[i];
         for (size_t j = 0; j < 3; ++j) {
              printf("%c", curr);
         }
         ++i;
     }

     printf("\n");
     return 0;
}
```

Task 5: Write 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(void) {
     char* numbers = "123 256 789";
     int to_add[3];
     sscanf(numbers, "%i %i %i",
          to_add, to_add + 1, to_add + 2);

     char s[32];
     sprintf(s, "The result is %d\n",
          to_add[0] + to_add[1] + to_add[2]);

     printf("%s", s);
     return 0;
}
```

Task 6: To define a **program** which defines two integer arrays, and then copy array1 values into array2, using function memcpy()
Note: give your array1 some initial values, for example:

int array1 [] = {1,2,3,4,5,6,7,8,9,0};
int array2[10];

[Hint: you may assume array1 and array2 have the same length]

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

int main(void) {
     int array1[] = {1,2,3,4,5,6,7,8,9,0};
     int array2[10];

     memcpy(array2, array1, sizeof(array1));
     return 0;
}
```

Task 7: Write a **function** which takes three char* (a, b, c) and then concatenate a and b to become a new string, put the string into c, you can assume both a and b have valid string and c is large enough. The prototype is as:

void my_concat (const char *a, const char *b, char* c);

```
void my_concat(const char* a, const char* b, char* c) {
    strcat(c, a);
    strcat(c, b);
}
```

Task 8: Write a **program** to compare each pair of the given strings and print the bigger one for each pair, using function **strcmp()**

(1) "Az" and "aZ"
(2) "ABCDEF" and "a"
(3) "sushi-roll" and "unagi"

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

#define SIZE 3

int main(void) {
    char* arr_1[SIZE] = {"Az", "ABCDEF", "sushi-roll"};
    char* arr_2[SIZE] = {"aZ", "a", "unagi"};

    for (size_t i = 0; i < SIZE; ++i) {
        printf("%s\n", strcmp(arr_1[i], arr_2[i]) > 0 ?
                    arr_1[i] : arr_2[i]
            );
    }
    return 0;
}
```

Task 9: To define a **program** to find the second ‘a’ in string “apple-pineapple”, using function **strchr**, print “yes” if you found the second, print “no” otherwise.

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

int main(void) {
     char a_p[] = "apple-pineapple";
     char* substr = strchr(a_p, 'a');
     size_t remaining_a = 2;

     while (substr != NULL && remaining_a > 0) {
          substr = strchr(substr, 'a');
          if (substr != NULL) {
               --remaining_a;
          }

          if (remaining_a == 0) {
               printf("yes\n");
               // distance in bytes
               printf("%ld\n", substr - a_p);
               return 0;
          }

          ++substr;
     }

     printf("no\n");
     return 0;
}
```

[Challenge: Try to see whether you can print the index of the second ‘a’, if found]

Task 10: Write a **function** which takes in two strings, and then prints the longer one, using strlen() to understand which one to print

```
void print_longer(const char* str_1, const char* str_2) {
     printf("%s\n",
          strlen(str_1) > strlen(str_2) ?
          str_1 : str_2
     );
}
```