CSDS 297/397 Special Topics: C/C++ Programming Spring 2021

Assignment 2: Functions, Arrays, & Pointers

Instructor: Orhan Özgüner Due: Sunday, January 24 before 11:59 PM

Please download `HW2.zip`. It contains project templates for each question, with a `main` and a header file containing the function prototypes you should use.

Question 1

Write a function with the following prototype:

```
void codeCipher(const int amt, char * str, int n)
```

This function should take a pointer to a sequence of characters `str` which is of size `n`, and shift the value of each character by `amt`. For instance, shifting the character `a` by 1 should produce `b`, and shifting `a` by 2 should produce `c`.

Shifts should loop within the uppercase and lowercase characters. If a character is shifted enough that it stops being in the original category, it should start again from the beginning of the category. For instance, shifting `z` by 1 should produce `a`, and shifting `Z` by 2 should produce `B`. All other characters should not be shifted.

Thus, if we call

```
codeCipher(5, "The answer to Question 1 is...!", 31)
```

the result would be

```
"Ymj fsxbjw yt Vzjxynts 1 nx...!"
```

Question 2

Write a function with the following prototype:

```
std::string buildSentence(char components[], int lc, int places[], int lp)
```

This function will build a string based on an array of characters and an array of index places. `lc` will always be the length of `components` and `lp` will always be the length of `places`. Each integer in `places` will be nonnegative and less than `lc` (so that the builder cannot exceed the length of `components`). The string is created by using the entries in `places` as indices to sample characters from `components`. For instance,

```
buildSentence("ol,wrd eH", 9, {8, 7, 1, 1, 0, 2, 6, 3, 0, 4, 1, 5}, 12)
```

would return `"Hello, world"`.

Question 3

Write a function with the following prototype:

```
std::string flipEveryXthWord(std::string sentence[], int size, int x)
```

This function takes in a `sentence` as an array of strings. There are a total of `size` strings. Each string can contain one or more words, separated by spaces. Your function should return a single string with every `x`th word of the strings in `sentence` reversed. For instance,

```
flipEveryXthWord({"This gnirts i", "s on", "t a test !drow"}, 3, 2)
```

should return `"This string is not a test word!"`.

Question 4

Write a function with the following prototype:

```
void moveD(int * block, int width, int height, int target)
```

This function should take in a pointer to a block of memory containing integers, which is `width` "wide" and `height` "tall". It should then find every instance of an integer equal to `target`, and should move this instance as far down as it can possibly go. The integer that was originally in the position the target now takes, should be placed where the target was originally. If it is not possible to shift the target, the function should skip it and move on. The block is altered "in place"; the function never needs to return anything whether it is successful or not. For instance, the following block

1	2	**0**	**0**	5
2	2	9	8	4
3	2	9	**0**	4
6	6	6	6	**0**

is called with

```
void moveD(&block, 5, 4, 0)
```

This would leave the block looking like this:

1	2	*6*	*6*	5
2	2	9	8	4
3	2	9	**0**	4
6	6	**0**	**0**	**0**