Lab 3: Functions

For this lab, create a new directory named `lab3` under your `cs449` directory and create your program there:

```
mkdir lab3
cd lab3
```

String Functions

For this lab, you are going to write a program named `lab3.c` which contains a series of functions related to string manipulations. Note that a string in C is simply an array or characters that ends with a null character (`'\0'`). **Note** that you are **NOT** allowed to use any predefine string functions from C library. The following are functions that you must implement listed by their signatures follow by their descriptions:

- `void strcpy(char *dest, const char *src)`

  Copies the string from `src` to `dest` including the null character.

- `void strcat(char *dest, const char *src)`

  Concatenates the string `src` to the end of the string `dest`.

- `void strrev(char *str)`

  Reverses the string `str`. For this function, there is no destination. Simply put the result in the original string itself.

- `void substring(char *dest, int start, int end, char *src)`

  Copies the substring from the string `src` from index `start` (inclusive) to index `end` (exclusive) to `dest`. For this function, you **MUST** handle the following unusual situations:

  - If `end < start`, the result is an empty string.
  - If `start < 0`, assuming that the `start` is 0.
  - If `end < 0`, assuming that the `end` is 0.
  - If `start` is greater than the length of the string `src`, the result is an empty string.
  - If `end` is greater than the length of the string, assuming that the `end` is the length of the string.

  Note that the result string must be a null-terminated string.

Example

Suppose you use the following `main()` function:

```
int main(void)
{
    char str1 = "Hello ";
    char str2 = "World!!!";
    char buffer1[100];
    char buffer2[100];

    strcpy(buffer1, str1);
    printf("%s\n", buffer1);
    strcat(buffer1, str2);
    printf("%s\n", buffer1);
    strrev(str2);
    printf("%s\n", str2);
    substring(buffer2, 0, 5, buffer1);
    printf("%s\n", buffer2);
    substring(buffer2, 6, 12, buffer1);
    printf("%s\n", buffer2);
}
```

The output of the above program should be:

```
Hello
Hello World!!!
!!!dlroW
Hello
World!
```

What to Hand In

First, let us go back up to our cs449 directory:

```
cd ..
```

Now, let us first make the archive. Type your username for the USERNAME part of the filename:

```
tar cvf USERNAME_lab3.tar lab3
```

And then we can compress it:

```
gzip USERNAME_lab3.tar
```

Which will produce a `USERNAME_lab3.tar.gz` file.

If you work on `cs449.cs.pitt.edu` (`thoth`) you can skip to the next section. **If you use your own machine, you need to transfer the file to `cs449.cs.pitt.edu` first**. This can simply be done by a command line. For example, assume that your username is `abc123` and you are in the same directory as the file `abc123_lab3.tar.gz`. To transfer the file to `cs449.cs.pitt.edu` use the following command:

```
scp abc123_lab3.tar.gz abc123@cs449.cs.pitt.edu:.
```

The above command will copy the file to your home directory in `cs449.cs.pitt.edu`. If you want to copy it to your `private` directory, use the following command:

```
scp abc123_lab3.tar.gz abc123@cs449.cs.pitt.edu:./private/.
```

Copy File to Submission Directory

We will then submit that file to the submission directory:

```
cp USERNAME_lab3.tar.gz /afs/cs.pitt.edu/public/incoming/CS0449/tkosiyat/sec1
```

Once a file is copied into that directory, you cannot change it, rename it, or delete it. If you make a mistake, resubmit a new file with slightly different name, being sure to include your username. For example `USERNAME_lab3_2.tar.gz`. **Check the due date of this lab in our CourseWeb under Labs/Recitations**.