Electrical & Computer Engineering Department, University of Maryland, College Park

Fall 2019 **ENEE140** Dr. Gang Qu

Project 4: Python to C Converter

Discussed: Tuesday November 19, 2019
Posted" Tuesday, December 3, 2019
Due: 11:59 pm Monday December 11, 2017

Project Objective:

1. learn the basics on user interface design and implementation
2. learn basic python commands
3. master skills for multi-dimensional array manipulation
4. get familiar with defining, implementing and using functions

Project Description:

In this project, you will design and implement a program to emulate usage of python methods. Python is a higher-level programming language than C in the sense that many useful functions are encapsulated within convenient commands that can simply be called. Your program should take in an input file in order to form a List. Then you will utilize a menu for the user to determine which python methods you will be performing on the List. These methods will be implemented as C functions by you.

General Requirements

1) Your program should run with the following command line arguments:

```
a.out   inputLength    input.txt      output.txt
```

Here, a.out is the compiled executable. InputLength is the number of lines in the input.txt file. Input.txt is a file that contains the information we will put into a 2D Array. Output.txt where the 2D Array's contents will be written to if a certain command is invoked.

You should check that the user has input the command line arguments and files correctly.

a. If the number of arguments is incorrect, print the following message and exit.

```
Incorrect number of arguments provided.
Usage: a.out inputLength input.txt output.txt
```

b. If any of the files cannot be opened, print the following error message and exit:

```
XXX.txt could not be opened properly.
```

Here XXX.txt is the name of the file that caused the error. If multiple files have an error than print an error message for each of them.

2) You will need to design a data structure that will hold all the contents of the input.txt file.

   a. Input.txt will hold random information on multiple different lines

   b. A 2D array will be needed to hold this information. This will ultimately be an array with each entry being an entire line within the input.txt file.

c. No line will be longer than 256 characters.

3) You will need to write a function for each of the different python methods we want to use to manipulate the information from the input.txt file. These are the prototypes and explanations for each function:

Please note that MAX_STRING is 256 and it is the length of the longest possible line from the input.txt file.

a. `int count(char List[][MAX_STRING],int index,char substring[]);`

This function takes in an array of strings and goes to the string specified by index. It then proceeds to find the number of occurrences of substring within this string specified by index.

For example:

String: "This is Kiran, he is a UTF."

Substring: "is"

"is" occurs three times within the string "This is Kiran, he is a UTF."

b. `void strip(char List[][MAX_STRING],int index);`

This function takes an array of strings and removes the leading and trailing space from the string at the specific index.

c. `void reverse(char List[][MAX_STRING],int size);`

This function takes in an array of strings and reverses the order of the strings in the array about the midpoint. Size is the number of strings within the 2D array.

For example:

2D Array holds:

"Kiran"

"Punit"

"Joseph"

"Hunter"

"Esa"

The 2D array will hold after the function call:

"Esa"

"Hunter"

"Joseph"

"Punit"

"Kiran"

d. `int listIndex(char List[][MAX_STRING],char s[],int size);`

This functions takes in an array of strings and returns the index where the specified string s occurs. Size is the number of strings within this array.

e. `void sort(char List[][MAX_STRING],int reverse,int size);`

This function takes in an array of strings and sorts it in alphabetical order. Reverse set equal to 1 is normal alphabetical order, but reverse set equal to 0 is reverse alphabetical order. Size is the number of elements within the array.

f. `int len(char List[][MAX_STRING],int index);`

This functions returns the number of characters of the string at the specified index position in the array of strings.

g. `int replace(char List[][MAX_STRING],char s1[],char s2[],int size);`

This function searches the array of strings for the existence of the string s1. If s1 exists, then it will be replaced by s2 and 1 will be returned. If s1 does not exist in the array of strings, then 0 will be returned. Size is the number of strings within the array.

h. `void delete(char List[][MAX_STRING],int size);`

This will set every position in the array of strings to '\0' Size is the number of strings in the array of strings.

i. `void print(char List[][MAX_STRING],int size);`

This will print out the current contents of the array of strings. Size is the number of strings within the 2D array.

j. `void writeFile(FILE* out,char filename[],char List[][MAX_LINE],int size);`

This function will take the file pointer out and open the file specified by filename. It will then write the contents of the array of strings to this output file. Size is the number of strings within the 2D array.

4) Once these functions have been created, they have to be implemented via user input. The user will decide through the use of an interface which functions are applied to the array of strings holding the information from input.txt. The user Interface will look like this:

```
Welcome to the Python Processor Program!

Below are a bunch of different Python Functionalities to
perform on the List of information you provided!

What would you like to do?

1) Python count method on String in List!

2) Python strip method on String in List!

3) Python reverse method on List!

4) Python index method on String in List!

5) Python sort method on List!

6) Python len method on String in List!

7) Python replace method on String in List!

8) Print out the current contents of the List!(Includes the
Line Number)
```

```
9) Python write method to print current contents of list to
the outputFile!

10) Python delete method List and then exit the program!

Choose an option(1-10)!
```

a. Once one of these options are chosen, you must perform the functionality of that option by using one of the functions implemented above.

b. If the user gives an incorrect input, an error message needs to be printed followed by this menu again:

   ```
   Invalid input! Try again.
   ```

c. The user has 3 chances to give an incorrect input. After the third incorrect attempt, the program should terminate.

d. After making a selection, you must then prompt the user for further information in order to make the function being tested work;

   For example, if Option 1 is chosen to use the count method. You will have already gotten a 2D array of the information from the file. Then, you must prompt a user for the index in the array that we want to analyze. You must also ask the user for the substring that you want to find within the string.

   These kinds of prompts may be done in any fashion you want as long as that you gather the information necessary to use the function that correlates with the choice.

Special Notes:

1. String.h and its functions may be used in this project.
2. A sample executable and some input files will be provided.

Project Requirements:

1. You **must** program using C under GLUE UNIX system and name your program **p4.c**.
2. Your program must be properly documented, particularly on the functions.
3. Submit your program **p4.c** electronically before the due time with the command below (replace the **? in 010?** by your own section number):

   submit 2019 fall enee 140 010? 3 p4.c

Grading Criteria:

Correctness:		87%
Function prototypes (assigned separately)		10%
Good coding style:		5%
Proper documentation:		10%
program that does not compile under GLUE:		-100%
late submission:	within the first 24 hours after due time	-40%
	after 24 hours of the due time:	-100%

TEMPLATE: You must use this template and follow the guidelines strictly.

```
/* Project 4 Template */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_LINE 256
/*=================================================
Function Prototypes
The parameter in size is the number of strings inside each list within each function
=================================================*/

//Takes in the list and an index within the list and returns the number of times substring occurs
within that line
int count(char List[][MAX_LINE],int index,char substring[]);

//Takes in the List and removes the leading and trailing whitespace from the element in the list at
index
void strip(char List[][MAX_LINE],int index);

//Takes in the List and reverses the strings about the midpoint of the List
void reverse(char List[][MAX_LINE],int size);

//Takes in a List and returns the index where the string s occurs.
int listIndex(char List[][MAX_LINE],char s[],int size);

//Takes in a List sorts the List in alphabetical order or backwards based on the parameter input.
void sort(char List[][MAX_LINE],int reverse,int size);

//Returns the length of a string at a specific index within the List
int len(char List[][MAX_LINE],int index);

//Print out the current contents of the List
void print(char List[][MAX_LINE],int size);

//Replaces String s1 with String s2 within the List. Returns 1 if replacement occurred or 0 if s1
doesn't exist.
int replace(char List[][MAX_LINE],char s1[],char s2[],int size);

//Empty Contents of the List and end the program.
void delete(char List[][MAX_LINE],int size);

//Print out the current contents of the List
void print(char List[][MAX_LINE],int size);

//Writes contents of List into output filename
void writeFile(FILE* out,char filename[],char List[][MAX_LINE],int size);
```

```
/*==================================================
Main Function
==================================================*/

int main(int argc, char *argv[]){
          /*==================================================
Read in all the Lines of Input file.
a. You are given the number of lines in this file.
b. You are given that no line will have more than 256 characters.
          ==================================================*/
if(argc != 4){
  printf("Incorrect number of arguments!\n Usage: a.out listLength input.txt output.txt");
                                                                          return 0;
                                          }
//Number of lines in input file
int lines=atoi(argv[1]);

//Array that will act as a python list to hold all the file information.
char stringList[lines][MAX_LINE];

FILE *inputFile=fopen(argv[2],"r");

return 0
}
```