

Introduction to Programming

CS1325 – Fall 2018

Project– Crossword

Individual Project– Crossword

November 01, 2018

Due Date: December 4, 2018 – 10:00pm (through eLearning)

Demo: No later than December 7th

Introduction

In this assignment you will write a C program. Your program should compile correctly and produce the specified output. *You will need to meet the grader to demo your project and answer questions about your code.*

Please note that the computer program should comply with the commenting and formatting rules as has been done in class. For example, there should be a header for the whole program that gives the author's name, class name, date, and description. End braces should be commented, and there are alignment and indenting requirements as discussed. Please ask if you have any questions.

Program #1

Given an input file (named `CrosswordInput.txt`) with the following format:

```
M,N,O,S,L,I,W,E,R,E,L,Y,T,L,E,A,G,N
A,H,O,O,V,E,R,T,A,Y,L,O,R,V,E,N,N,A
D,F,D,R,O,O,S,E,V,E,L,T,O,N,O,M,I,M
I,N,T,P,M,H,I,E,G,D,I,L,O,O,C,O,D,U
S,O,N,L,I,J,Q,A,D,A,M,S,S,R,N,N,R,R
O,X,L,O,G,E,F,F,M,O,I,R,E,E,G,R,A,T
N,I,B,T,S,O,R,A,O,R,M,O,V,A,W,O,H,N
F,N,H,U,R,R,B,C,R,N,L,R,E,G,B,E,W,E
R,O,S,D,C,O,E,A,E,I,O,N,L,A,U,A,J,R
K,W,U,N,L,H,H,F,N,I,O,S,T,N,S,W,A,U
E,R,B,A,J,B,A,C,F,S,S,M,N,H,H,R,D,B
N,E,W,L,O,T,O,N,K,E,C,E,I,H,T,H,A,N
N,T,H,E,H,L,A,C,A,K,J,N,N,H,O,I,M,A
E,R,G,V,N,C,A,F,I,N,G,H,U,H,A,J,S,V
D,A,R,E,S,J,C,N,T,T,A,R,N,B,O,Y,A,E
Y,C,A,L,O,D,L,N,O,S,I,R,R,A,H,W,E,O
N,E,N,C,N,E,T,N,N,O,T,N,I,L,C,O,E,S
D,A,T,Y,Y,P,O,L,K,G,A,R,F,I,E,L,D,R
```

- A) Display a menu with the following options: **(5 points)**
- Enter file location (full path)
 - Display the crossword
 - Find a word
 - Exit
- B) When option “a” is chosen:
- Read a c-string from the user **(5 points)**
 - Write a program to read the file content and determine the array dimensions(i.e. char crossword[x][y]: find the values for x and y that should be used as upper bound for any loop)**(5 points)**
The size is always nxm, where $n, m \leq 80$
 - Load the file content into a two dimensional array, where each cell is a character distinct of ‘,’ , ‘\0’, ‘ ‘ or ‘\n’(i.e. only letters are valid) **(10 points)**
 - If the file does not exist, or it contains invalid characters, display an error message and display the main menu again. **(10 points)**
- C) When option “b” is chosen, display the file/array content using the following format: **(5 points)**

```

M N O S L I W E R E L Y T L E A G N
A H O O V E R T A Y L O R V E N N A
D F D R O O S E V E L T O N O M I M
I N T P M H I E G D I L O O C O D U
S O N L I J Q A D A M S S R N N R R
O X L O G E F F M O I R E E G R A T
N I B T S O R A O R M O V A W O H N
F N H U R R B C R N L R E G B E W E
R O S D C O E A E I O N L A U A J R
K W U N L H H F N I O S T N S W A U
E R B A J B A C F S S M N H H R D B
N E W L O T O N K E C E I H T H A N
N T H E H L A C A K J N N H O I M A
E R G V N C A F I N G H U H A J S V
D A R E S J C N T T A R N B O Y A E
Y C A L O D L N O S I R R A H W E O
N E N C N E T N N O T N I L C O E S
D A T Y Y P O L K G A R F I E L D R

```

- D) When option “c” is chosen: ask the user for a word and determine whether the word was found or not (i.e. print a message like “%s found!” or “%s was not found”). **(40 points)**
 Assume that valid words are always:
- Left to right – right to left or,
 - Top to bottom - bottom up or,
 - Any combination @ 45 degrees (i.e. from left(top) to bottom(right))
- If the word is found, use a structure to store: the word and the row/col coordinates **(10 points)**
- E) When option “d” is chosen, exit the program.

Deliverables

You will turn in two files, a Text file and a C source code file.

The text file should contain:

- 1) Instructions to compile and use your program
- 2) Whether it could find or not the following words using the input file provided by the instructor: **(10 points)**
 - a. KENNEDY
 - b. ROOSEVELT
 - c. PIERCE
 - d. CLINTON
 - e. LINCOLN

The C source code file should:

- 1) Comply with all of the formatting requirements already discussed.
- 2) Display the output