Lab 4

All questions in this lab should be written with C.

Question 1:

In your computer, the order of computing a complicated expression is managed by Reversed Polish Notation. Reversed Polish Notation has the property that you can always process from it left to right. For example, in Reversed Polish Notation:

a+d*(b-c) is represented by a,d,b,c,-,*,+.

You can refer to the following link to learn more about Reversed Polish Notation: https://en.wikipedia.org/wiki/Reverse_Polish_notation

Given a in Reversed Polish Notation, your computer performs the following task:

1. From left to right, store each number it meets in a sequence.

2. If it encounters an operation symbol, take the last two numbers in the sequence and calculate the result. Write the result at the end of the sequence.

Your task in this question is:

1. deal with Reversed Polish Notation input and output the result.

Sample input: 1,2,3,4,-,*,+ (Input is split by comma)

output: -1

2. convert normal input into Reversed Polish Notation input

Sample input: 1+2*(3-4)

output: 1,2,3,4,-,*,+ (Output is split by comma)

Question 2:

In this question, you will write a game called hangman. In this game, player will guess an unknown word. At the beginning of the game, player will get a word whose every letter is displayed by an underline. And player will have a score 10 at the beginning. Each time, player should input a letter, if it appears in the word, the corresponding underlines will be replaced by that letter. If the letter doesn't appear in the word, the player will lose 1 point. When the score finally turn to 0, the game is over and the player loses the game. When all letters of the word are guessed, the player wins.

We will provide a word bank on canvas containing about 100 words. You can write these words in your program. Each time you should randomly pick one word to let the player guess. When testing your program in lab session, we will specify a certain word to use, so make sure you also have a line of code that allow you to specify a certain word.

To make the game more interesting, there are some magic items player can use in this game. At the beginning of the game, the player has each of the following items.

"*": Show one letter in the word randomly. Note that if the word contains two or more same letters, they will not all be shown.

e.g. _ a _ _ _ _ _ (hangman).

"/": Randomly show the player two wrong letters.

"#": The player suicide and lost the game.

"@": Show the places where the letter is a vowel and replace the underlines by an "@". e.g. _@ng_@_ (hangman).

"$": Use an item randomly.

OUTPUT:

Each time the output contains five rows.

The first row: The score of the player.

The second row: The items player has.

The third row: The word displayed.

The fourth row: The known wrong letters.

The fifth row: “Please enter a letter (a-z):”

“You win.” if the whole word is displayed

“You lose.” if the score becomes 0, or item “#” is used. Then output the right answer.

Sample: (hangman)

Score: 10

Item: * / # @ $

_ _ _ _ _ _ _

Already known wrong answers:

Please enter a letter (a-z): e

Score: 9

Item: * / # @ $

_ _ _ _ _ _ _

Already known wrong answers: e

Please enter a letter (a-z): @

Score: 9

Item: * / # $

_ @ _ _ _ @ _

Already known wrong answers: e

Please enter a letter (a-z): /

Score: 9

Item: * # $

_ @ _ _ _ @ _

Already known wrong answers: e, f, t

Please enter a letter (a-z): n

Score: 9

Item: * # $

_ @ _ _ _ @ n

Already known wrong answers: e, f, t

Please enter a letter (a-z): a

Score: 9

Item: * # $

_ a n _ _ a n

Already known wrong answers: e, f, t

Please enter a letter (a-z): m

Score: 9

Item: * # $

_ a n _ m a n

Already known wrong answers: e, f, t

Please enter a letter (a-z): *

Score: 9

Item: # $

h a n _ m a n

Already known wrong answers: e, f, t

Please enter a letter (a-z): g

Score: 9

Item: # $

h a n g m a n

Already known wrong answers: e, f, t

You win.