In The Name Of God

The Most Compassionate And Merciful

Problem Set -6

Prefix, Infix, Postfix

------------------------------------------------

Sheet info :

* Problem Set 6
* Due Date : Not Set
* Just upload scorable questions on HWS.
* File name format : “**studentNumber\_PS6.zip**”
* Do not hesitate to ask any question from your graders!

# **1) Write a function to evaluate postfix expressions.**

***Input:*** *21-3+*

***Output:*** *4*

***Input:*** *45-32\*+1+*

***Output:*** *6*

# **2) Write a function to evaluate prefix expressions.**

***Input:*** *+-213*

***Output:*** *4*

***Input:*** *++-45\*321*

***Output:*** *6*

# 

# **3) Write a function to evaluate infix expressions.**

***Input:*** *2-1+3*

***Output:*** *4*

***Input:*** *4-5+(3\*2)+1*

***Output:*** *6*

# **4) Write a function to convert infix to postfix and calculate the phrase.**

***Input:****a+b\*c*

***Output:*** *abc\*+*

***Input:*** *str = “a+b\*c-(d/e+f^g^h)”*

***Output:*** *abc\*+de/fgh^^+-*

# **5) Write a function to convert infix to prefix and calculate the phrase.**

***Input:*** a+b\*(c^d-e)^(f+g\*h)-i

***Output:***abcd^e-fgh\*+^\*+i-

# **6) Write a function to convert prefix to infix.**

***Input:*** +\*abc

***Output:***((a\*b)+c)

# **7) Write a function to convert postfix to infix.**

***Input:*** ab\*c+

***Output:***((a\*b)+c)

# **8) Write a function to evaluate infix boolean expressions.**

***Input:*** A = 1 // (True) // global

B = 0 // (False) // global

C = 0 // (False) // global

(~B | A) & C // passing to the function

***Output:***0

# **9) Write a function to evaluate prefix boolean expressions.**

***Input:*** A = 1 // global

B = 0 // global

C = 0 // global

&|~BAC // passing to the function

***Output:***0

# **10) Write a function to convert infix boolean expressions to prefix**

***Input:*** (~B | A) & C

***Output:***&|~BAC

**11) Write a function to determine whether an expression’s bracketing, is valid or not.**

Function declaration:

int is\_valid (string)

Example:  
char e[] = “(3 + ((5) \* -7))”

is\_valid (e);  
return value: non-zero

char e[] = “(3 (+) ())”

is\_valid (e);  
return value: 0

**12) Sort the numbers of stack using recursion functions.**

# 

# **13) Reverse a string using stack.**

**14) Write a program using as many functions u can which shows all the permutations possible for the given string.**

Be careful about the characters which are repeated.

input: aba

aab

baa

aba

Input: abc

abc

acb

bac

bca

cab

cba

**15) Write a program in which u are Given an array of integers and k, find the max for each k members in the array.(use stack for this function)**

Input :

arr[] = {1, 2, 3, 1, 4, 5, 2, 3, 6}

k = 3

Output :

3 3 4 5 5 5 6