COMP-2310 DATA STRUCTURES AND ALGORITHMS

ASSIGNMENT-2

OCTOBER 17,2022

1.Write a new expression similar to the following (Change the positions of orerators):

E1 = a+b\*c+(d\*e+f)\*g [ Use your own operators : /,\*,+…]

E2 = a-b\*c/(d\*e+f)-g

a)Convert your expression to postfix form.

E2 (Postfix form): abc\*de\*f+/-g-

b) Trace the algorithm Postfix on slide 39 using your expression.

You can use the format on slide 49.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Current symbol | Operator Stack | Postfix String |
| 1 | a |  | a |
| 2 | - | - | a |
| 3 | b | - | ab |
| 4 | \* | - \* | ab |
| 5 | c | - \* | abc |
| 6 | / | - / | abc\* |
| 7 | ( | - / ( | abc\* |
| 8 | d | - / ( | abc\*d |
| 9 | \* | - / ( \* | abc\*d |
| 10 | e | - / ( \* | abc\*de |
| 11 | + | - / ( + | abc\*de\* |
| 12 | f | - / ( + | abc\*de\*f |
| 13 | ) | - / | abc\*de\*f+ |
| 14 | - | - | abc\*de\*f+/- |
| 15 | g | - | abc\*de\*f+/-g |
| 16 | Empty the stack (final step) |  | abc\*de\*f+/-g- |

2.Write a new expression similar to the following :

E2= 6\*((5+(2+3)\*8)+3)

E3=6-((2\*(8-3)/5)+1)

1. Convert your expression to postfix form.

E3 (Postfix Form) = 6283-\*5/1+-

1. Trace EvalPostfix algorithm on slide 54 using your expression.

| Symbol | **Stack** |
| --- | --- |
| 6 | 6 |
| 2 | 6 2 |
| 8 | 6 2 8 |
| 3 | 6 2 8 3 |
| - | 6 2 5 |
| \* | 6 10 |
| 5 | 6 10 5 |
| / | 6 2 |
| 1 | 6 2 1 |
| + | 6 3 |
| - | 3 |

Result: 3

3.Write a program that receives some input values and

a) Displays these values by pointers,

1. Displays their memory addresses.

#include <iostream>

int main() {

int num1, num2;

std::cout << "num1: ";

std::cin >> num1;

std::cout << "num2: ";

std::cin >> num2;

int \*pNum1, \*pNum2;

pNum1 = &num1;

std::cout << pNum1 << std::endl;

std::cout << &num1 << std::endl;

std::cout << \*pNum1 << std::endl;

std::cout << num1 << std::endl;

pNum2 = &num2;

std::cout << pNum2 << std::endl;

std::cout << &num2 << std::endl;

std::cout << \*pNum2 << std::endl;

std::cout << num2 << std::endl;

}

4.Write an algorihm in pseudocode

a) to insert a new node inside a linked list (Study slide 25)

insertFirst(*node*) {  
 *node*->next = head;  
 head = *node*;

}

b)to delete a node from inside a linked list (Study slide 28)

deleteFirst() {

temp = head;

head = head→next;

delete temp;

return head;

}