National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Data Structure**	**Course Code:**	
	Program:	**BSCS**	**Semester:**	**4th**
	Name:		**Section:**	**4A**
	Registration #:		**Assessment**	**Quiz04**
	Due Date:	**16th June, 2020**	**Time Duration:**	**60 mins**

Instruction/Notes:

1. Late submissions will not be entertained.

Q1: [5]

Given the code already provided to you to construct an expression tree (code is uploaded), modify the code that accepts an arithmetic expression written in prefix (Polish) notation, builds an expression tree, and then traverses the tree to evaluate the expression. The evaluation should start after a complete expression has been entered. For example given input **+3*52** the output should be **13**

Q2: [5]

Consider a hash table consisting of M = 11 slots, and suppose nonnegative integer key values are hashed into the table using the hash function h1():

```
int h1 (int key) {
 int x = (key + 7) * (key + 7);
 x = x / 16;
 x = x + key;
 x = x % 11;
 return x;
}
```

Suppose that collisions are resolved by using linear probing show the result of inserting 43,23,1,0,15,31,4,7,11,13

Q3: [5]

The keys 24, 39, 31, 46, 48, 34, 19, 5, and 29 are inserted (in the order given) into an initially empty AVL tree. Show the AVL tree after each insertion.