CS 442/542 Fall 2020. Quiz 7 **Name:______________________________**

Implement a lex and yacc file that determines if an input file is syntactically correct based on the grammar shown below. You can assume the input comes from stdin (i.e. you are not using IOMngr or the macro I gave you for the lex file). Your program should output the string "Success" when the input is correct and it should output "Failure" when the input is not correct.

Below I gave you a template for the lex and yacc files. There are not semantic routines (except printing Success) to write. You only use the parsing features of yacc (but you will have to put empty actions for each production except the action of printing "Success".

To submit your answer upload one pdf file that contains your implementation of the lex file and the yacc file.

Here is the grammar

Goal -> { List }
List -> NestedList List
List -> ε
NestedList -> (NestedList List)
NestedList -> (ItemList)
ItemList -> Ident, ItemList
ItemList -> IntLit, ItemList
ItemList -> Ident
ItemList -> IntLit

An Ident is a sequence of one or more letters and an IntLit is a sequence of one or more digits. Examples of legal strings in the language are

{(10)}

{(10,20,abc)}

{(10)(20)(30)}

{((10))}

{((10,20)(abd))}

{((10, 20)(abc,def,100)(2,4,8,16))(1,2,3)((9,99)(8,88))}

lex template

```
%{
   #include "y.tab.h"
%}

letter  [A-Za-z]
digit   [0-9]

%%
```
Your work goes here
```
%%

int yywrap() {
   return 1;
}
```

yacc template

```
%{
extern int yylex();
extern int yyerror(char *);
#include <stdio.h>
#include <stdlib.h>
%}

%token Ident
%token IntLit

%%
```
Your work goes here
```
%%

int yyerror(char *s) {
```
Your work goes here
```
}

int main(int argc, char * argv[]) {
   yyparse();
}
```