Readme for Assignment 4

1. Group Members

Chen Jiali (3035085695) Cao Chao (3035111949)

2.Features Implemented

In this assignment, we have implemented the following features.

1). Array (1D/2D/3D)

- Declaration:

eg. array arr1[10]; //global 1D array

```
func()
{
   array @arr2[2][3]; //global 2D array
   array arr3[4][5][6]; //local 3D array
}
```

2). Assignment:

eg. arr1[2] = 6; // int assignment

```
@arr2[1][1] = 'a'; // char assignment
arr3[1][2][3] = "abc"; //string assignment
```

3). Pass by reference as function parameters:

eg. fun(array a[10], array b[2][3], array c[4][5][6])

```
{
   a[3] = 6;
   putc(b[1][1]); // 'a'
   puts(c[1][2][3]); // "abc"
}
fun(arr1, arr2, arr3);
```

4). String Manipulation

eg. array str[20] = "Hello world";

```
str[0] = 'h'; // the string becomes "hello world"
```

3. Usage Instruction

1). Download/Clone this repository into a suitable place. (Linux/Mac)

2). Install **flex** and **bison** if you don't have them yet.

3). Compile C6C

make c6c

4). Compile NAS

make nas

5). Compile test case into assemble code (Test cases can be found in the [tests](./tests) folder)

./c6c test.sc > test.as

6). Run test case

./nas test.as

4. Demo Case

We demo our C6C compiler using an application for playing the 24-point game written in the C-like language.

4.1 Compile the game

./c6c 24pts.sc > 24pts.as

./nas 24pts.as

4.2 Playing instructions

1. Press 's' to start the game.
2. Choose a problem from the problem set (0 - 49).
3. Enter your calculation, press Enter to submit.
4. Or, give up and press 'h' check for the solutions.
5. The program will check if your solution is correct and count your score accordingly.

* Following are the screen shots for the major steps in game playing.

```
posnet@c3234-Ubuntu-1404:~/Documents/COMP3235$ ./nas play.as
please press 's' to start and any other key to exit:
s
please input question no (0~49): 10
==========================
A       2       5       3
==========================
[please notice: 0 represents 10]

Please input your calculation, or press 'h' to checkout the solutions.
[Please use 'A', '0', 'J', 'Q', 'K' to denote 1, 10, 11, 12, 13]
[You can use '+', '-', '*', '/' and '()' for calculation]
[Press Enter to submit.]
h
This problem can be solved as:
((A+2)+5)*3
(A+(2+5))*3
(A+2)*(5+3)
please press 's' to start and any other key to exit:
```

Figure 1: Screen shot for case user pressing 'h' for help.

```
please press 's' to start and any other key to exit:
s
please input question no (0~49): 10
==========================
A       2       5       3
==========================
[please notice: 0 represents 10]

Please input your calculation, or press 'h' to checkout the solutions.
[Please use 'A', '0', 'J', 'Q', 'K' to denote 1, 10, 11, 12, 13]
[You can use '+', '-', '*', '/' and '()' for calculation]
[Press Enter to submit.]
((A+2)+5)*3
computation result:
24
Congradulations!
Your current score: 1
please press 's' to start and any other key to exit:
```

Figure 2: Case for user enters the answer and the answer is evaluated to be correct.

```
please press 's' to start and any other key to exit:
s
please input question no (0~49): 10
==========================
A       2       5       3
==========================
[please notice: 0 represents 10]

Please input your calculation, or press 'h' to checkout the solutions.
[Please use 'A', '0', 'J', 'Q', 'K' to denote 1, 10, 11, 12, 13]
[You can use '+', '-', '*', '/' and '()' for calculation]
[Press Enter to submit.]
Q+Q
computation result:
24
Sorry, the answer is not valid for the current question.
Your current score: 1
please press 's' to start and any other key to exit:
```

Figure 3: Case for user enters the answer and the answer is evaluated to be invalid.

```
please press 's' to start and any other key to exit:
s
please input question no (0~49): 10
==========================
A       2       5       3
==========================
[please notice: 0 represents 10]

Please input your calculation, or press 'h' to checkout the solutions.
[Please use 'A', '0', 'J', 'Q', 'K' to denote 1, 10, 11, 12, 13]
[You can use '+', '-', '*', '/' and '()' for calculation]
[Press Enter to submit.]
A+2+5+3
computation result:
11
Sorry, the answer is not corret.
Your current score: 1
please press 's' to start and any other key to exit:
```

Figure 4: Case for user enters the answer and the answer is evaluated to be wrong.

4.3 Features of the demo case

1). Generation of a problem set: we first generate 50 solvable problems and store them in a global 2D array Q.

2). Implementation of a calculator: we implement a parser and evaluator for infix expressions to handle user's input:

*For the calculator, we implement two stacks using two 1D arrays: the value stack and the opertor stack.

We take the user's input as a string containing values (A, 2 ~ 10, J, Q, K) and operators '(',')' < '+','-' < '', '\\'.

*The input will be scaned once, and the values and operators will be pushed into stacks accordingly.

*For operators, if the operator stack is empty, or contains only a '(', or the current operator has a higher precedence than the one on stack top, then the operator will be pushed onto the stack.

*otherwise, the top of the two stacks will be computed until a ')' is met or the above conditions is not satisfied.

*when the scanning finishes, the remaining part of the two stack will be computed.

*the last number in the value stack is the result.

3). Implementation of Stack operations: push, pop, peek using 1D array.

4). Implementation of a function to generate the 24 points solutions: used when the user asks for help and when to generate the problem set.

5). Usage of global and local variables: eg. the problem set is defined globally, which requires reading and writing operations at differnet places of the program.

6). String implemented as char arrays: eg. string ALLTYPE and ALLOP are treat as char arrays.

7). Access global varaible in function. e.g. checkAns().

8). Pass array as function parameters: fetchQ() takes a 2d array as parameter, the array is passed by reference.