

Homework 5 . Subroutines. Functional Programming.

Submit your solution to Blackboard by 11:59pm of Tue Apr 21 .

1. List a) the name(s) of people who have contributed to your solution of this homework, and b) their contribution (briefly). If you worked by yourself, the answer of this question would be "N.A." Note the answer of this question is worth of 5% of this homework.

2.

```
int m=10;
foo(m);
print(m); // print the value of m to screen
```

```
proc foo(int a) {
    a := a+5
}
```

What would be the respective result of the `print` if the parameter `a` of `foo` is passed by

- (a) value,
- (b) reference,
- (c) value-result, and
- (d) result.

3. Consider the following program where `foo` uses pass by value for `a`:

```
int m=10;
m = foo(3);
print(m);

int foo(int a) {
    int x;
    if (a == 1)
        x = 1;
    else x = foo(a-1) + a;
    return x;
}
```

- (a) Which type of memory is used for global variable `m` and local variable `x` and parameter `a`?
 - (b) Trace the execution of the program by drawing the memory layout before and after each call of `f00`. Each of your layouts should show clearly where all variables are located. You can cook up the returning addresses but they should be consistent.
4. Using **common lisp** (on installation and its use see Appendix), write a program for each of the following task and submit a text file of your program to Blackboard with name "Q4.lisp". *You programs are expected to be executable and correct.*
 - (a) Using lisp write a function `max` whose input is three numbers and the output is a maximal of the three. For example, `(max 1 2 3)` will be evaluated as 3.
 - (b) Using lisp, write the function `f00` whose input is a number x and a list L and output is true if x occurs in L , false otherwise. When you solve the problem, practice the top down approach with the method of divide and conquer (or decomposition) that was discussed in the class. For example, `(f00 1 '(1 2 3))` is evaluated to true while `(f00 6 '(1 2 3))` false.
 - (c) (Optional) Given the definition of functions (in the downloadable file beside this homework document), write the following functions: `isFinal(X)` returns true if X is a final state and false (nil) otherwise. `getNextState(s, c)` returns the next state given a state s and a symbol c . `scan(s, input)` where s is a state and $input$ is a list of symbols. It outputs, to the console, all states, which the automata has gone through in terms of $input$. It further outputs to console ACCEPT if $input$ is accepted by the automata and REJECT Otherwise. For example, `scan(q0, (0, 1))` will output Q0 Q2 Q3 REJECT.

Appendix. Information on common lisp.

1. Lisp, its interpreter and installation, editor
 - (a) The language we use is Common Lisp.
 - (b) The interpreter we use is CLISP.
 - For Windows system, refer to the section of "Download and install CLISP" at <https://bit.ly/3c0MaH5>
 - For other operating systems, go to <http://www.clisp.org/> and search "Get CLISP" and find instructions for different OS.
 - (c) It is an excellent idea to use an editor (e.g., notepad++/emacs) that can tell the matching of the parenthesis which will be of great help.
 - (d) You may refer to the link (type the address, but DON'T click it) <http://cs.gmu.edu/~sean/lisp/LispTutorial.html> for a quick understanding of clisp.
2. How to run your program.

Under command prompt of windows, assume you have the file `wholeProgram.clisp` in your current directory. After typing `clisp -i wholeProgram.clisp`, you will be

in the clisp interpreter with the program `wholeProgram.clisp` loaded. Under clisp prompt, you can test the functions defined in your program by typing, e.g., `(max 1 2 3)`.