I2P(II)2022 Yang Written Exam Practice Part I

1. Explain the following terms and their usages
(a) register
(b) CPU (central processing unit)
(c) RAM (memory)
(d) compiler
(e) assembler
(f) linker
(g) loader

2. Give the flow how your .c program is executed by a computer.

3. Explain the relations of (a) variables (b) values (c) memory address (d) register.

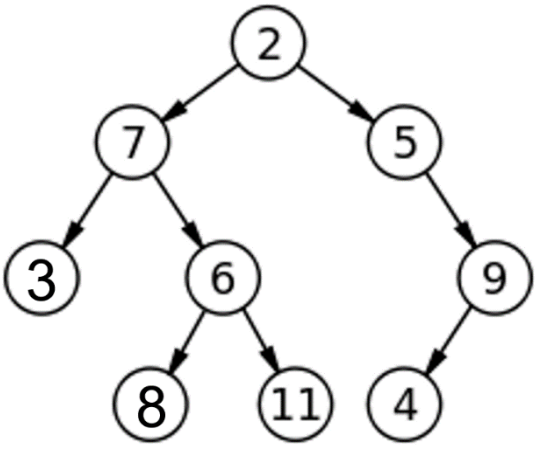
4. Explain the relation of (a) computer system (b) logic gate (c) integrated circuits.

5. Explain how to use AND, OR, NOT, XOR gates to implement a one-bit adder.

6. Explain the relation of (a) Boolean operations (b) logic gates (c) bitwise operations

7. Explain the purposes of the three steps “lexical analysis”, “parsing process” and “code generation” in the compilation process.

8. Show the pre-order, in-order, and post-order traversal sequences of the following binary tree.



9. Given the pre-order and in-order traversal sequences of a binary tree:

Pre-order: 2 7 3 6 8 11 5 9 4
In-order: 3 7 8 6 11 2 5 4 9

show the structure of this binary tree.

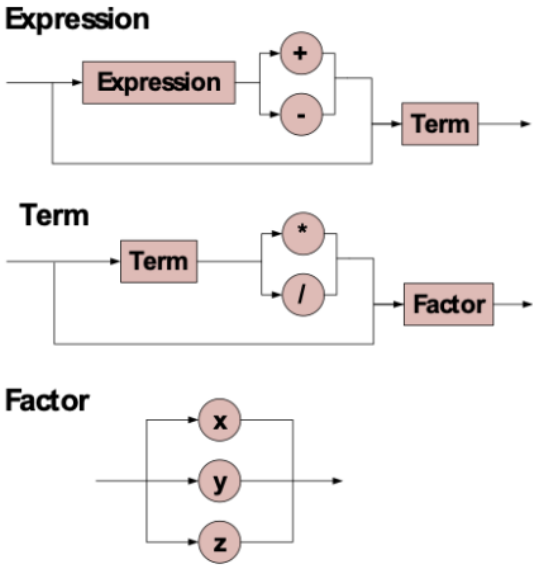
Mini Project 1&2

1. Draw the parse tree / syntax tree of the 2 inputs:
   (i) a = x + y * z
   (ii) a = x + y + z

2. Algebraic expressions manipulating variables x, y and z, such as “x-y*z+x/y”, can be described by the following grammar recursively:

   Expression := Term | Expression ADDSUB Term
   Term := Factor | Term MULDIV Factor
   Factor := x | y | z

   where “|” means “or”, or equivalently by the following syntax diagrams



   For the string **x + y * z**, draw the parse tree based on the above syntax diagrams.

3. A palindrome is a string that reads the same forward and backward, such as otto or madamimadam. To make things simple, we shall consider describing only the palindromes with alphabet {0, 1}. This language includes strings like 0110, and 11011, but not 011 or 0101. Design a syntax diagram representing the grammatical structure of the palindromes with alphabet {0, 1}.

4. How can we reduce the clock cycles of the generated assembly code? Please provide two potential methods and briefly describe them.

5. Can we pre-calculate the results (values of x, y, z) directly for all inputs? Assuming we want the final value of x, y, z to be at r0, r1, r2.
   For example:
   x = 1
   y = 2
   z = x + y + 3
   x = z - y
   can be easily simplified to (x,y,z) = (4,2,6), that is
   MOV r0 4
   MOV r1 2
   MOV r2 6
   EXIT 0

   If we can pre-calculate all the required results and reduce them into only 4 instructions, why bother generating the parser/syntax tree?

6. What is the final value of the registers (r0, r1) of the following instructions?
   MOV r1 3
   MOV r0 r1
   EXIT 0

7. Please write out any assembly code that can represent following operation (assuming x, y, z values are at r0, r1, r2 initially and we want the final values also be at r0, r1, r2):
   a = x + y * z
   x = a

8. Can we use for-loop to achieve the same compile result as using recursion?

9. Briefly describe the most difficult problem you encountered in mini-project-1.

10. Briefly describe the most difficult problem you encountered in mini-project-2.