1. Given the following infix expression: **(a + b + c) \* d - e \* f \* g** and the precedence and associativity of the operators in the expression are like those in C. Transverse it into the Polish prefix format?

🡪 - \* + + a b c d \* \* e f g

2. Determine each of the following statements is TRUE or FALSE?

- Cambridge Polish Prefix expression requires the number of operands of every operator fixed.

a. True **b. False**

*As Cambridge Polish Prefix expression has ( and ), it allows the number of* *operands of every operator not fixed. For example, (\* a b) or (\* a b c d) are correct.*

- Polish prefix expression is suitable just for binary operators

a. True **b. False**

- Normal prefix expression has operators staying outside ().

**a. True** b. False

3. Assume that the precedence and associativity of operators in the following infix expression are like those in C: a \* b \* ( c - e - f) \* g. Rewrite the expression in other notations while keeping the appearance order of operands. No space is allowed.

The Polish prefix notation of the above expression is

**\*\*\*ab--cefg**

The Cambridge Polish prefix notation of the above expression is

**(\*ab(-cef)g)**

The Polish postfix of the above expression is

**ab\*ce-f-\*g\***

4. Given the following expression in Polish prefix notation: **\* \* a + b c - d \* e f**

With the minimum of ( ), the same appearance order of operands and no space, the equivalent expression in infix notation is 🡪 **a\*(b+c)\*(d-e\*f)**

5. Assume that the value of variable a is 6. Select possible values of the following C expression: a + (a = 3) \* a? 🡪 **24, 12, 15, 21.**

6. Assume that the value of variable x is 5. After the execution of the following logical expression written in C (where logical expressions are short-circuit evaluated): ((x = 1) == 1) && ((x = 2) == 3) && ((x = 3) == 3), what is the value of x ? 🡪 **2**

*In short-circuit evaluation, operand evaluation order is defined from left to right, so, when the second operand (x=2)==3 is evaluated to false, the execution is terminated and x will have the value of 2.*

7. When execution an assignment statement, the left hand side expression will be calculated to determine its 🡪 **Address.**

8. When execution an assignment statement, the right hand side expression will be calculated to determine its 🡪 **Value.**