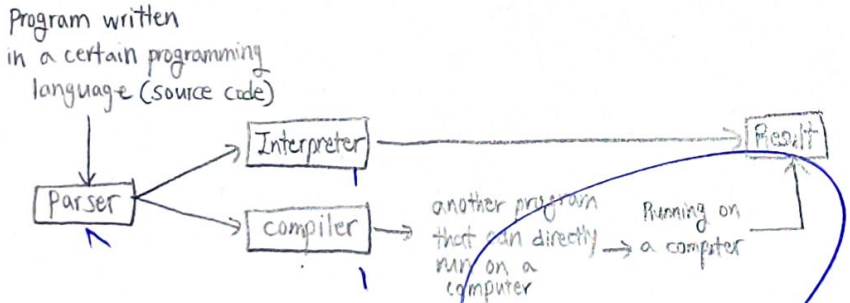
1. Write a complete BNF for the FOURAE programming language which supports four fundamental arithmetic expressions. (1)

For concrete syntax, we use +, -, *, / for addition, subtraction, multiplication, division operators respectively. Based on examples, fill the blank.

Examples	BNF
{/ 2 {- -4 5}} {+ 1 2.4} {* {+ 2.0 -3} 1} {- 4 -5.1} 3.141529 -0.567 -123.234 {+ {- 1 2} {+ 3 4}}	<FOURAE> ::= <number> | {- <FOURAE> <FOURAE>} | {+ <FOURAE> <FOURAE>} | {/ <FOURAE> <FOURAE>} | {* <FOURAE> <FOURAE>} <number> ::= -<digit>+<below_dot>? | <digit>+<below_dot>? <below_dot> ::= .<digit>+ <digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Note: ? (an item is optional), + (an item exists 1 or more times), '-' followed by digits, e.g., -4, means a negative sign but not an operator for subtraction. Answers must be complete (no partial points).

2. Draw a figure how source code is interpreted to produce a result. Then, briefly describe each component in your figure. (1)



⇒ At first, source code is written in a concrete syntax. Then parser checks if the code is valid based on syntax and converts it into an abstract syntax. And then, interpreter which is a program converts the code in abstract syntax into corresponding number, producing a result directly. Unlike the interpreter, compiler needs another program so that the code is converted into machine code. When it can run on a computer, it produces a result.

3. Why do most programming languages support 'substitution'? Answer shortly. (1)

When there is a repeated expression, a programmer can make some mistakes. Also, evaluating the expression wastes computational cycle. Therefore, most programming languages implemented substitution with the idea of identifier which identifies the value of an expression

* Are you going to submit the optional HW2? (Yes) / No

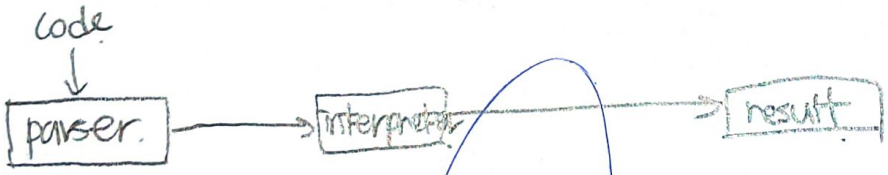
1. Write a complete BNF for the FOURAE programming language which supports four fundamental arithmetic expressions. (1)

For concrete syntax, we use +, -, *, / for addition, subtraction, multiplication, division operators respectively. Based on examples, fill the blank.

Examples	BNF
{/ 2 {- -4 5}} {+ 1 2.4} {* {+ 2.0 -3} 1} {- 4 -5.1} 3.141529 -0.567 -123.234 {+ {- 1 2} {+ 3 4}}	<FOURAE> ::= <number> | {/ <FOURAE> <FOURAE>} | {* <FOURAE> <FOURAE>} | {+ <FOURAE> <FOURAE>} | {- <FOURAE> <FOURAE>} <number> ::= -? <digit>+ <below_dot>? <below_dot> ::= .<digit>+ <digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Note: ? (an item is optional), + (an item exists 1 or more times), '-' followed by digits, e.g., -4, means a negative sign but not an operator for subtraction. Answers must be complete (no partial points).

2. Draw a figure how source code is interpreted to produce a result. Then, briefly describe each component in your figure. (1)



The code written by a user is converted to abstract syntax by a parser. The parser finds the patterns which is already defined without knowing the meaning and changes it to abstract syntax checking if the code is valid based on syntax. Then, an interpreter understands the abstract syntax semantically and produces result.

3. Why do most programming languages support 'substitution'? Answer shortly. (1)

It is because writing repeated expressions might make a programmer make a mistake as well as a computer waste computational cycles.

* Are you going to submit the optional HW2? Yes | No