Sl. No	Problem Statement

	Assignment 9 (20 March 2024) **Total =5+5= 10 marks**
	
	Implementation of shift reduce parser :
	
	Shift-reduce parsing is a form of bottom-up parsing in which a stack holds grammar symbols and an input buffer holds the rest of the string to be parsed. $ is used to mark the bottom of the stack and also the right end of the input. Initially, the stack is empty, and the string w is on the input, as follows:
	
	           STACK **INPUT**
	              $** **w$
	
	During a left-to-right scan of the input string, the parser shifts zero or more input symbols onto the stack, until it is ready to reduce a string of grammar symbols on top of the stack. It then reduces to the head of the appropriate production. The parser repeats this cycle until it has detected an error or until the stack contains the start symbol and the input is empty:
	
	           STACK **INPUT**
	           $S **$**
	
	Upon entering this configuration, the parser halts and announces successful completion of parsing. There are actually four possible actions a shift-reduce parser can make:
	
	    Shift : This involves moving symbols from the input buffer onto the stack.
	
	    Reduce : If the handle appears on top of the stack then, its reduction by using appropriate production rule is done i.e. RHS of a production rule is popped out of a stack and LHS of a production rule is pushed onto the stack.
	
	    Accept : If only the start symbol is present in the stack and the input buffer is empty then, the parsing action is called accept. When accepted action is obtained, it means successful parsing is done.
	
	    Error : This is the situation in which the parser can neither perform shift action nor reduce action and not even accept action.

EXAMPLE :

Production Rules :

E → E + E | E ✳ E

E → id | (E)

E → E / E

Input :

id1+id2*id3

Output :

Grammar are

E → E + E | E ✳ E

E → id | (E)

E → E / E

STACK	INPUT	ACTION
$	$id_1 * id_2$ $	shift
$ id_1	$* id_2$ $	reduce by $F \rightarrow id$
$ F	$* id_2$ $	reduce by $T \rightarrow F$
$ T	$* id_2$ $	shift
$ $T *$	id_2 $	shift
$ $T * id_2$	$	reduce by $F \rightarrow id$
$ $T * F$	$	reduce by $T \rightarrow T * F$
$ T	$	reduce by $E \rightarrow T$
$ E	$	accept

String id1+id2*id3 is accepted

Implement the following using the shift-reduce parser :

Roll_No: 2110342211IT001 - 2110972211IT021

Production Rules :

$$M \rightarrow N \,|\, M+M \,|\, M*M \,|\, (\,M\,)$$

$$N \rightarrow x \,|\, y \,|\, Nx \,|\, Ny \,|\, N5 \,|\, N6$$

Input : 1) (x656 * y6)

2) N55 + (x6 * y)

Roll_No:2110491211IT022 - 2110491211IT041

Production Rules :

$$P \rightarrow Q\;R\;;$$

$$R \rightarrow int$$

$$R \rightarrow float$$

$$S \rightarrow R\,,\,id$$

$$S \rightarrow id$$

$$S \rightarrow (P)$$

Input : 1) int id , id ;

2) float id R , id ;

Roll_No: 2110491211IT042 - 2110491211IT063

Production Rules :

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow id$$

$$E \rightarrow (E)$$

$$E \rightarrow a \,|\, b$$

$$E \rightarrow E \,/\, E$$

Input : 1) ((id+id) *id)/id   2) if(a*b)

Roll_No:2110491211IT064 - 2110491211IT085

Production Rules :

$$S \rightarrow S+S$$

$$S \rightarrow S*S$$

$$S \rightarrow (S)$$

$$S \rightarrow x$$

$$S \rightarrow y$$

Input : 1) (x+x)*a

2) +x*y+(x)