1. Consider the following grammar:

$$S \rightarrow cAt$$
$$A \rightarrow a \mid \epsilon$$

(a) [5] Calculate FIRST and FOLLOW for the nonterminals S and A

(b) [5] Compute its LL(1) parse table

(c) [5]Show the behavior of the parser on the sentence ca.

2. Consider the following grammar G:

$$E \rightarrow E + E$$
$$E \rightarrow E * E$$
$$E \rightarrow (E)$$
$$E \rightarrow id$$

(a) [10] Is G LL(1)? If yes, show its LL(1) parse table. If not, rewrite the grammar so that it is LL(1) and then show its parse table.

(b) [5] Show the behavior of the parser on the sentence $id + id * id$.

Note: Operator $*$ takes precedence over operator $+$, i.e. $* \cdot> +$.

3. [30] Consider the following grammar G:

$$S' \rightarrow S$$
$$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$$
$$A \rightarrow d$$
$$B \rightarrow d$$

(a) Is G SLR(1)? If yes, give the parse table. Otherwise, show why.

(b) Is G LR(1)? If yes, give the parse table. Otherwise, show why.

(c) Is G LALR(1)? If yes, give the parse table. Otherwise, show why.

4. [10] Consider the following grammar G for expressions:

$$E \rightarrow E + E \mid E * E \mid id$$

and the operator-precedence relations are given in the following table:

	id	$+$	$*$	$\$$
id		$\cdot>$	$\cdot>$	$\cdot>$
$+$	$<\cdot$	$\cdot>$	$<\cdot$	$\cdot>$
$*$	$<\cdot$	$\cdot>$	$\cdot>$	$\cdot>$
$\$$	$<\cdot$	$<\cdot$	$<\cdot$	

Show the process of paring the sentence $id + id * id$ by an operator-precedence parser.

5. Translate the expression $a := -(a + b) * (c + d) + (a + -c)$ into

   (a) [5] a syntax tree

   (b) [5] postfix notation

   (c) [5] three-address code

6. The **repeat** statement has the following form

   repeat S **until**(E)

   In other words, S will be executed until the condition E is true.

   (a) [10] Write down the actions of the grammar to translate the **repeat** statement into three-address code.

   (b) [5] Translate the statement

   ```
   i := 1;
   repeat
       i := i + 1;
   until (i >= 10)
   ```

   into three-address code.

   Hint: *E.true* and *E.false* store the labels to which control flows if E is *true* and *false*, respectively. Generated codes for S and E are stored in attributes *S.code* and *E.code*, respectively.