SANKALCHAND PATEL COLLEGE OF ENGINEERING, VISNAGAR
COMPUTER ENGINEERING DEPARTMENT
B.E. Semester – VI (Computer Engineering)

ASSIGNMENT - 1

Subject: System Programming (160706) **Date: 07/03/2014**

Q.1	List various phases of a language processor. Explain roles of first two phases of it. Also explain symbol table.
Q.2	Define following terms: 1. Execution Gap 2. Interpreters 3. Non Terminal Symbol 4. Derivation 5. Reduction 6. Parse Tree
Q.3	Write unambiguous production rules (grammar) for arithmetic expression containing +, – , *, / and ^ (exponentiation).Construct parse tree and abstract syntax tree for: **<id> – <id> * <id> ^ <id> + <id>**
Q.4	Explain Left Recursion, Left Factoring and Backtracking in Top-down parsing with suitable example.
Q.5	Explain working of LL(1) parser. Parse the following string. **|- <id> * <id> * <id> + <id> -|**
Q.6	Given a grammar, E → TA A → +TA | ε T → VB B → *VB | ε V → id | (E) Develop an LL(1) parser table and parse the string: **id * (id + id)**
Q.7	Write unambiguous production rules to produce arithmetic expression consisting of +, -, *, /, ^ (exponent), id. Use them for parsing **id ^ id ^ id * id + id / id** using shift-reduce parser (Naïve bottom up parsing). Also list limitation(s) of the method.
Q.8	What is operator precedence parsing? Show operator precedence matrix for following operators: +, - , * , (,). Parse the string: **|- <id> + <id> * <id> -|**
Q.9	Consider following grammar **S -> aSbS | bSaS | epsilon** Derive the string abab. Draw corresponding parse tree. Are these rules ambiguous? Justify.
Q.10	Construct DFA for following regular expression. (a* | b*) a* ab#
Q.11	Given the grammer, perform the top-down parsing for the string: **+*35*45** E = +TE | E T = *VT | V V = 0 | 1 |…| 9
Q.12	What is bottom up parser? Explain operator precedence parser. Let a grammar for a language is E → E+E | E*E | id. Check validity of following string using stack based operator precedence parser. **id * id + id * id**