**G.H. Patel College of Engineering and Technology**

**Department of Computer Engineering**

**2170701– Compiler Design, AY: 2018-19**

**Assignment 3**

Subject Code: 2170701 Subject Name: Compiler Design

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | | --- | |  | |  | |
| 1 | |  |  | | --- | --- | | Convert the following statement into triple, indirect triple and quadruple forms.  1)A= (B+C) $ E + (B+C) \*F  2) - ( a \* b ) + ( c + d ) - ( a + b + c + d )  3) – (a\*b)+(c\*d)+(a\*b\*c)  4) a = (a \* b + c) ^ (b + c) \* b + c.  5) a=b\*-c+b\*-c |  | |
| 2 | What is intermediate code? What is its importance? Discuss various representations of three address code. |
|  |  |
| 4 | Draw syntax tree and DAG for following statement. Write three address codes from both.  a = (a + b \* c) ^ (b \* c) + b \* c ^ a |
| 5 | |  | | --- | | Explain how type checking & error reporting is performed in compiler. | |
| 6 | |  | | --- | | Translate the arithmetic expression a\*-(b+c) into  1. Syntax tree  2. Postfix notation  3. Three address code | |
|  |  |
| 7 | Explain the following:  1) The Handle  2) Directed Acyclic Graph  4) Conflicts in LR Parsing  5) Parser Generator  6) Dependency Graph  7) Locality of reference |
| 8 | Show that the following grammer  S->Aa | bAc | dc | bda  A->d  is LALR(1) but not SLR(1). |
|  |  |
|  | Submission Deadline : on or before **20/09/2019** |