DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER VI [INFORMATION TECHNOLOGY]
SUBJECT: (IT 608) LANGUAGE TRANSLATOR

Examination	**: First Sessional**	**Seat No.**	**:**
Date	**: 06/01/2014**	**Day**	**: Monday**
Time	**: 12.45 to 2.00**	**Max. Marks**	**: 36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.
5. Here | is rule separator and ^ stands for NULL.

Q.1 Do as directed.

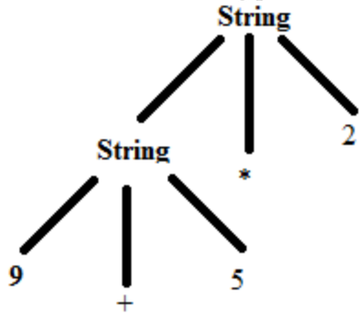
(a) Parsing is the process of determining **[1]**
- (A) String of terminals that can be generated by grammar.
- (B) String of non terminals that can be generated by grammar.
- (C) String of tokens that can be generated by grammar.
- (D) None of these.

(b) Which of the following suffice to convert an arbitrary CFG to LL (1) Grammar? **[1]**
- (A) Removing left recursion
- (B) Factoring the grammar
- (C) Removing left recursion and Factoring the grammar
- (D) None of these

(c) Consider the following issues of lexical analyzer and choose correct one: **[1]**
- I. Simplify the phase
- II. Compiler efficiency is improved
- III. Compiler produces target code faster
- IV. Compiler portability is enhanced

(A) I,II,III (B) I, III, IV (C) I,II,IV (D) All of these

(d) Consider the following parse tree. **[1]**



Which of the following statement is true?
(A) Both + and * is having equal precedence
(B) + is having higher precedence over *
(C) * is having higher precedence over +
(D)None of these

(e) Consider following grammar : S→cAd A→cb | ac | a **[1]**
For input string 'cad', How many times the brute-force will backtrack?
(A) 5 (B) 4 (C) 3 (D) 2

(f) What is back-patching? How to overcome this situation. State clearly. **[1]**

(g) Match the following phases of compiler with the tasks that they do. **[2]**

(i)Semantic analyzer	(I) intermediate code translated to machine code
(ii)Code generator	(II)determine meaning of source program
(iii)Code optimizer	(III)determine tokens in the source program
(iv)Lexical analyzer	(IV) produce more efficient object program.

(h) Consider following grammar: S → aSbS | bSaS | ^. Is the grammar ambiguous? If Yes then give example to prove it. **[2]**

(i) Justify: It is not necessary that the load time address of program is same as the translation time address. **[2]**

Q.2 Attempt ***Any Two*** from the following questions. **[12]**

(a) For the following 'C' fragment, identify and list the lexemes that make up tokens. [6]

```
#define int char
main()
{   int i=65;            // i is initialized
    printf("sizeof(i)=%d",sizeof(i));
}
```

(b) Remove the left recursion or factor the following grammar (if needed). And find FIRST and FOLLOW set. E → T+E | T, T → int | int * T | (E) [6]

(c) Draw transition diagram for the following rules of **Language L1**. [6]
Whitespaces- blanks, tabs, newlines. **Keywords**- if, else, then, end, start
User identifiers- start with capital letters, followed by one or more letters.
Comments- start with **<!** - - Anything that is comment - - >
Character constants- anything between ` `. Apostrophe can be included in the character constant, but it should occur in pair. Ex- `it``s valid statement`.

Q.3 (a) Determine following grammar is LL(1) or not. **[8]**

S→qABC A→a | bbD B→a | ^ **C**→b | ^ D→ c | ^

If it is LL (1), draw parse table. And using the table show the parsing process for string '**qbbca**'. Else explain the reason why it is not LL (1).

(b) Write RDP for the following grammar : S → cS | bA, A → d | CcA **[4]**

OR

Q.3 (a) Determine following grammar is LL (1) or not. **[8]**

S→A, A→aB | d, B→bBC | f, C→g

If it is LL (1), draw parse table. And using the table show the parsing process for String 'abbfgg'. Else explain the reason why it is not LL (1).

(b) Give scanner algorithm for the rules of the given **Language L1** in Q.2-C **[4]**