BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(END SEMESTER EXAMINATION)

CLASS: BE **SEMESTER : VI**
BRANCH: CSE **SESSION : SP/19**

SUBJECT: CS6105 COMPILER DESIGN
TIME: 3:00 HOURS **FULL MARKS: 60**

INSTRUCTIONS:
1. The question paper contains 7 questions each of 12 marks and total 84 marks.
2. Candidates may attempt any 5 questions maximum of 60 marks.
3. The missing data, if any, may be assumed suitably.
4. Before attempting the question paper, be sure that you have got the correct question paper.
5. Tables/Data hand book/Graph paper etc. to be supplied to the candidates in the examination hall.

Q.1(a) What are the features of a good language and a good compiler? [2]

Q.1(b) Discuss the challenges and application of compiler design [4]

Q.1(c) Explain the various phases of a compiler with help of the following program. [6]

```
main(){
int a, b;
float c,d;
c=a+b*10; d=a-b/2.0
}
```

Q.2(a) Which of the following expressions have l-values and / or R-values. [2]
(i) A[I+1] (ii) * A (iii) & A (iv) &(* A) (v) *(& A) (vi) * (&(&A))

Q.2(b) Explain each of the following parameter passing mechanism with suitable examples [4]
i) call-by-value ii) call-by-reference iii) call-by-name iv) copy-restore

Q.2(c) Write a Lex program that will captured the vowel in the text and replace into Uppercase. [6]

Q.3(a) What are steps to eliminate left recursion? [2]

Q.3(b) Explain the operator precedence parsing algorithm [4]

Q.3(c) Write a YACC program for the given production: [6]
E->E+E| E-E| E*E| E/E| (E) |-E| id

Q.4(a) Explain Recursive Descent Parser with suitable example. [2]

Q.4(b) Consider the grammar [4]
E-> BA
A->&BA|€
B->true|false
Show that the grammar is LL(1) and construct the predictive parsing table.

Q.4(c) Consider the grammar [6]

S --> Aa|bAc|Bc|bBa
A -->d
B -->d

Show the grammar is LR (1) but not LALR (1).

Q.5(a) Differentiate between the Abstract Syntax Tree and the Directed Acyclic Graphs with suitable example. [2]

Q.5(b) Consider the following grammar [4]
D->TL
T->int|float
L->L_1 , id| id
Write corresponding semantic action for each of the Non-Terminal and then construct the annotated parse tree for the input expression float a,b,c

PTO

Q.5(c) Generate the three address code for assignment statements, where the identifier may be array elements instead of simple variables. [6]

```
S -> L:=E
E->E+E | (E) | L
L->Elist]|id
Elist-> Elist, E| id[E
```

Using the above semantic action, draw the annotated parse tree for the statement

X[I,j]:=Y[i+j,k]-Z[k,l]

Q.6(a) Differentiate between SDD and SDT. Also, discuss different types of SDT's with simple examples. [2]

Q.6(b) What do you mean by runtime storage allocation? Explain the difference between static and dynamic allocations. [4]

Q.6(c) Consider the following piece of code: [6]

```
begin
    While a> b do
      begin
          if x>y then x=y+z
          else
               x=y-z
       end
```

Show the annotated parse tree and the intermediate code generation process.

Q.7(a) What are themes behind optimization techniques? [2]

Q.7(b) Explain the following with suitable examples: [4]

(i) Loop Fission (ii) Loop Interchange (iii)Loop Reversal (iv) Loop Splitting

Q.7(c) i)Consider the following C code segment. [6]

```
for (i = 0; i < n; i++){     for (j = 0; j < n; j++){
          if (i%2)  {x + = (4 * j + 5 * i);  y + = (7 + 4 * j);  } }}
```

Modify the program using suitable optimizing techniques

ii)Specify the necessary and sufficient conditions for performing loop optimization and dead code elimination. Give suitable examples.

::::::03/05/2019::::::E