VIT

Reg. No. : 21BCE1461

Final Assessment Test (FAT) - July/August 2023

Programme	**B.Tech.**	Semester	**Fall Inter Semester 22-23**
Course Title	**COMPILER DESIGN**	Course Code	**BCSE307L**
Faculty Name	**Prof. Sivakumar P**	Slot	**F1+TF1**
		Class Nbr	**CH2022232500858**
Time	**3 Hours**	Max. Marks	**100**

Section 1 (10 X 10 Marks)
Answer All questions

01. Demonstrate the different stages involved in the process of compilation **(5 Marks)**, and display the results generated by each stage for the given input: **(5 marks)** **[10]**

```
int a, b;
a= 10, b= 20;
if (a == b)
printf("both are equal");
else if(a>b)
printf("%d is greater", a);
else
printf("%d is greater, b);
return 0;
}
```

02. Consider the grammar G with the following productions. **[10]**

S → AB |eDa
A → ab |c
B → dC
C → eC | ε
D → fD | ε

Construct LL(1) parsing table for G. **(6 marks)**
Show the stack status, input and shift/reduce action used for parsing the string "efcf" **(4 marks)**

03. Consider the grammar G with the following productions. **[10]**

S → CC
C → cC
C → d

Construct Look Ahead LR (LALR) parsing table for G and analyze the different types of conflicts. **(7 marks)**
Show the stack status, input and shift/reduce action used for parsing the string "ccdd" **(3 marks)**

04. Construct the parse tree for the Boolean expression (a > b AND b < c OR d < e) **(5 marks)** and show the corresponding semantic translations. **(5 marks)** **[10]**

05. Construct the parse tree **(5 Marks)** and the corresponding three address code for the following program **(5 Marks).** **[10]**

```
switch(i-j)
{
case 1: X= a-b;
break;
case 2: Y= c+d;
break;
default: Z= e+f;
}
```

06. Derive at least minimum 3 strings from the following grammar G **(3 marks)** [10]
Grammar G: **(E->E+T | T, T-> T*F |F, F->(E) |id)** and translate the expression into
(i) Abstract syntax tree **(2 Marks)**
(ii) Quadruple representation **(2 Marks)**
(iii) Triple representation **(2 Marks)**
(iv) Indirect triple representation **(1 mark)**

07. Construct the basic block and flow graph from the following code block **(5 Marks).** Also identify the loops present in the flow graph **(5 Marks).** [10]

```
function findMax(M, n):
 m = 0
 v = 0
while v <= n:
    r = v
    s = 0
while r < n:
        x = M[r]
        s = s + x
           if s >= m:
             m = s
            r = r + 1
             v = v + 1
 return m
```

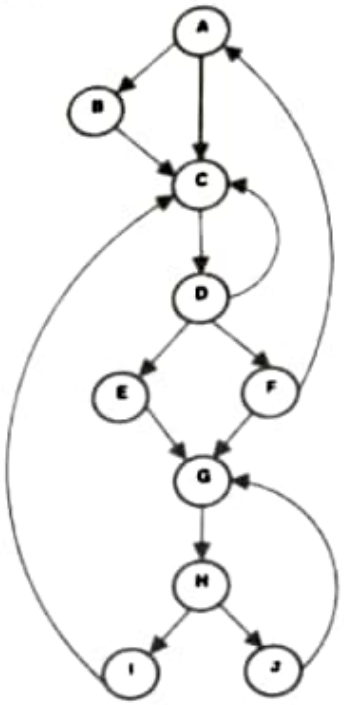
08. (i) Construct the Directed Acyclic Graph (DAG) for the following three address code. **(4 Marks)** [10]

```
T1 =a+b
T2 = a-b
T3= T1 *T2
T4= T1- T3
T5 = T4+T3
```

(ii) Apply back-patching in the generation of three address code for the following programming construct. Illustrate the back-patching through annotated parse tree **(6 marks)**

```
while a<b do
   If c<d then
x:= y+z
     else
x:= y-z
```

09. (i) Construct dominator tree and find natural loops in the below CFG. **(5 Marks)** [10]



(ii) Design code generator for the below sequence of three address code. **(5 Marks)**

t1 = a-b

t2 = a+c

t3= t1 - t2

p =t3+t2

10. Compare different parallel compiler machines with respect to instruction streams **(5 Marks)** and architecture **(5 Marks)**. Illustrate with relevant examples. [10]