ID NO.	

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY, BENGALURU

SCHOOL OF ENGINEERING

Weightage: 40 %   Max Marks: 80   Max Time: 02 Hrs.   14 May 2018, Monday

ENDTERM FINAL EXAMINATION MAY 2018

Even Semester 2017-18   Course: **CSE 217 COMPILER DESIGN**   VI Semester CSE

Instructions:

(i) Read the question properly and answer accordingly.

(ii) Question paper consists of 3 parts.

Part A

(3 Q x 8 M = 24 Marks)

Q1. For the given program, apply the following optimizations **in the same order.**

```
a := x *8
t1=a
b := 3
c := x
t1=c
d := b+c
t2=1
t3=t2+2
e := b * 2
f := b + c + t3
g := e * f
```

a) Algebraic optimization or strength reduction
b) Copy propagation
c) Constant folding
d) Common sub-expression elimination
e) Dead code elimination

Q2. Consider the following program.

```
for (i=2; i<=n; i++)
        a[i] = TRUE;
count = 0;
s = sqrt (n);
for (i=2; i<=s; i++)
        if (a[i])
        {       count++;
                for (j=2*i; j<=n; j = j+1)
                        a[j] = FALSE;
        }
```

a) Translate the program into three address code

b) Identify all basic blocks in your three address code and Build the flow graph for the three address code.

Q3. Consider the following program written in C language.

```
int function1(int abcdefghijklmn       int opqurstuvwxyz)
{
float mmlkjij=0;
if(abcdefghijklmn    != opqurstuvwxyz)
{ mmlkjij= mmlkjij+ abcdefghijklmn;}
else if (abcdefghijklmn<= opqurstuvwxyz) {
mmlkjij= mmlkjij+ opqurstuvwxyz;
}
elseif (abcdefghijklmn>= opqurstuvwxyz)
{ mml@kjij= mmlkjij* opqurstuvwxyz; }
```

Assume you are planning to change the existing Lexer and

a) Discuss the requirement of designing of lexical analyzer with respect to the above program.
b) Generate the possible valid and invalid tokens for this program

Part B

(1 Q x 16 M = 16 Marks)

Q4. Consider the grammar given below:

S→dA|Ab,   A→bA|c,   B→bB|c

a) Construct Push down automata.
b) Construct LR(0) Parsing Table.
c) State whether the given grammar is LR(0) or not. Justify your answer.

Part C

(1 Q x 40 M = 40 Marks)

Q5. Consider the program given below:

```
i= 10;
j=i+14;
sum=0;
while(i<j){
 um=sum+a*b+d;
  i=i+4;
  j++;}
```

With reference to the grammar given below perform the following:

<Stmts>→ <Stmt> | <Stmt>;<Stmts>| ε

<Stmt>→<E> | <Loop>

<E>→ <E>+<E> | < E>*<E> | <E> < <E> | <E> > <E> | <E> = <E> | <id>

<Loop>→ while(<E>){ <Stmts>}

<id>→ [a-z]$^{+}$ | [A-Z]$^{+}$

a) Construct the Parse tree.
b) Construct the AST
c) Generate the Three Address Code
d) Construct the Data Flow Graph by Identifying the Basic Blocks.
e) Perform all possible optimization in the basic blocks.
f) Perform the Liveness analysis
g) Construct the Interference Graph
h) Perform register allocation
i) Generate the Target code for x86 machine

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

ID NO:

PRESIDENCY UNIVERSITY, BENGALURU

SCHOOL OF ENGINEERING

Weightage: 20%   Max Marks: 40   Max Time: 1 hr.   2 April 2018, Monday

TEST – 2   **SET B**

Even Semester 2017-18   Course: **CSE 217 Compiler Design**   VI Sem. CSE

Instruction:

(i) Read the question properly and answer accordingly.
(ii) Question paper consists of 3 parts.
(iii) Scientific and Non-programmable calculators are permitted

Part A

(2 Q x 5 M = 10 Marks)

Q1. Consider the states of LR(1) parser

State0 A→• a, b A→a•, c B→ a•, b	State 1 A→• a, c A→a•, b B→ a•, a
State 2 A→• a, a A→•a, b B→ a•, b	State 3 A→• a, a B→•a, b

Can any states be merged to form LALR(1) parser? Justify.

Q2. Consider the program given below

```
1.     int main()
2.     {
3.     int p,a;
4.     for(int i=0;i<p; )
5.     {
6.      int a,m,n;
7.     a=a*i;
8.     i++;
9.     }
10.    a=100;
11.    }
```

Complete the symbol table given below which is a sorted array implementation.

Name	Characteristic Class	Scope	Declared	Referenced	Other information

Part B

(1Q x 12 M = 12 Marks)

Q3. Consider the Grammar E→E+E / E*E / (E) / a / b / c / d and the string belong to the grammar (a+b+c+d)*(c+d)+(a+b).

i. Give the AST and corresponding Three Address code.

ii. Give the DAG and corresponding Three Address Code.

Part C

(1Q x 18 M = 18 Marks)

Q4. Construct the LALR(1) parse table for the grammar given below.

S -> L = R

R -> L

L -> * R

L -> id

ID NO:

PRESIDENCY UNIVERSITY, BENGALURU

SCHOOL OF ENGINEERING

Weightage: 20 % Max Marks: 40 Max Time: 1 hr. 23 February Friday 2018

TEST – 1

Even Semester 2017-18 Course: **CSE 217 Compiler Design** VI Sem. **CSE**

Instruction:

(i) *Read the question properly and answer accordingly.*

(ii) *Question paper consists of 3 parts.*

Part A

(3 Q x 5 M = 15 Marks)

Consider the following C program to answer Q1, Q2 and Q3.

```
int main( )

{

 int intnum1, intnum2;

 scanf(“%d %d”,&intnum1,intnum2);

 if(intnum1<=intnum2)

 printf(“ I am in PresidencyUniversity,
Bengaluru”);

 else if(intnum1== intnum2)

printf(“ I am on IndustryVisit”);

}
```

1. Explain “Principle of longest match” in context of lexical analysis using the program given above.

2. A input buffer of size 10 is used for lexical analysis. Explain with the help of program given above, why the buffer size will have serious impact on compilation time.

3. Two input buffers of size 5 each is used for lexical analysis. Explain with the help of program given above, how the compilation time will improve by using two buffer.

Part B

(1Q x 10 M = 10 Marks)

4. Consider the following grammar G.
   Where, Non terminals are{ Email, Name} and Terminals are{ @, . , id}

   Email → Name @ **id . id**
   Name → **id | id .** Name

   a) Is this LL(1) grammar? Justify
   b) Modify the grammar in G to make it LL(1) if required.
   c) Construct the LL(1) parse table for the modified grammar.
   d) Derive the string **id.id@id.id**

Part C

(1Q x 15 M = 15 Marks)

5. Consider the following two grammars G1 and G2.

   G1: A→ aA /a   **G2:** A→ Aa /a

   a) Which one is not suitable for LR(0) parsing? Justify
   b) Are they SLR? Justify.