APJ ABDULKALAM TECHNOLOGICAL UNIVERSITY
08 PALAKKAD CLUSTER

Q. P. Code : 2A181 **(Pages: 4)** **Name** .....

Reg. No: .....

SECOND SEMESTER M.TECH. DEGREE EXAMINATION, APRIL 2018

Branch: CSE **Specialization: Computer Science and Engineering**

08 CS 6012 ADVANCED COMPILER DESIGN

Time:3 hours **Max. Marks: 60**

Answer all six questions.

Modules 1 to 6: Part 'a' of each question is compulsory and answer either part 'b' or part 'c' of each question.

Q.no.	Module 1	Marks
1.a	Obtain an annotated parse tree for the following syntax directed definition with synthesised and inherited attributes for the grammar for type declarations.	3

Production	Semantic rule
D→T L	L.in :=T.type
T→int	T.type:=integer
T→real	T.type:=real
L→ L1,id	L1.id:=L.in;addtype (id.entry,L.in)
L→id	addtype(id.entry,L.in)

Answer b or c

b	Construct a Syntax-Directed Translation scheme that translates arithmetic expressions from infix into postfix notation. Your solution should include the context-free grammar, the semantic attributes for each of the grammar symbols, and semantic rules. Show the application of your scheme to the input "3*4+5*2".	6
c	Explain parser stack implementation of Postfix SDT using an example of desk calculator.	6

Module 2

2.a	Consider following definition	3

D →T L	L.in := T.type
T →real	T.type := real
T →int	T.type := int
L →L_1 ,I	L_1 .in :=L.in; I.in=L.in
L →I	I.in = L.in
I →I_1 [num]	I_1 .in=array(numeral, I.in)
I →id	addtype(id.entry,I.in)

Obtain the parse tree and dependency graph for the string **int x[3], y[5]**

Answer b or c

b Obtain three address codes for the following statement and explain. 6

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

c Consider grammar and rules given below for array address translation and generating address code for array references: 3 6

E ⟶ E1 + E2	{E.addr = newtemp(); gen(E.addr '=' E1.addr '+' E2.addr);}
E ⟶ E1 * E2	{E.addr = newtemp(); gen(E.addr '=' E1.addr '*' E2.addr);
E ⟶ id	{E.addr = id.lexeme; }
E ⟶ L	{ E.addr = newtemp(); gen(E.addr '=' L.array.basename '['L.addr ']'); }
L ⟶ id [E]	{L.array = id.lexeme; L.type = L.array.typeofelement; L.addr=newtemp(); gen(L.addr '=' E.addr '*' L.type.width);}
L ⟶ L1 [E]	{L.array = L1.array; L.type=L1.type.typeofelement; t = newtemp(); L.addr = newtemp(); gen(t '=' E.addr '*' L.type.width); gen(L.addr '=' L1.addr '+' t); }

Function newtemp() returns a new temporary name
L.array.basename means name of the array
L.array.typeofelement means type of the element of the array
L.type.width means width of L.type

Assume size of integer to be 4 bytes, and lower bound of the arrays to be 0
Let A, B and C be 3X4, 4X5, and 3X5 arrays of integers respectively. Let i, j, and k be integers.
Construct an annotated parse tree for the expression C[i][j] + A[i][k]*B[k][j] and show the 3-address code sequence generated for the expression.

Q.no.	Module 3	Marks
3.a	What is activation record?	3

Answer b or c

b Write the algorithm for Mark-and-Compact Garbage Collector and explain. 6

c Explain the use of access links for finding non local data. Also explain the display implementation. 6

Q.no.	Module 4	Marks
4.a	Write the techniques for the optimization of basic blocks.	3

Answer b or c

b Apply simple code generation algorithm to generate code for the following statements. 6

t1=a-b
t2=c+d
c=e*t2
d=t1+t2

Show the instructions generated and changes in address and register descriptors while generating code each statement. Assume there are only three registers .t1 and t2 are temporary variables.

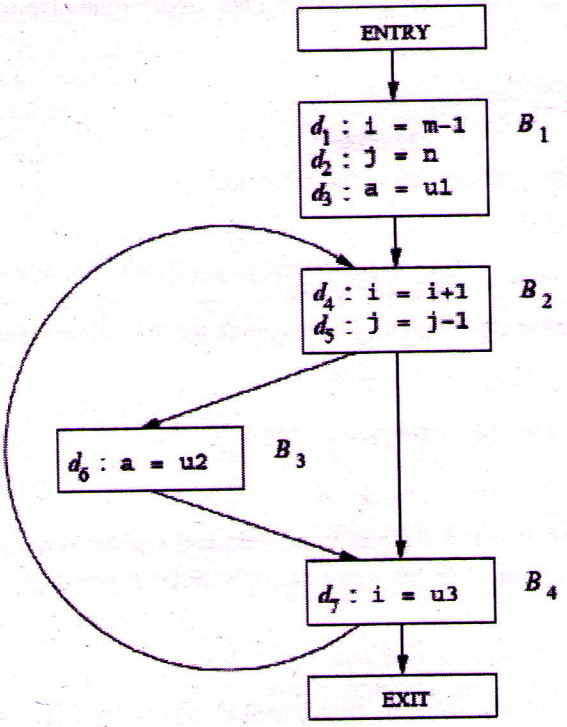
c Generate code for the following statements from labelled expression tree using Ershov numbers. t,u,v, and w are temporary variables . Assume the number of registers available is three. 6

t= a-b
u=c+d
v=e*u
w=t+v

Q.no.	Module 5	Marks
5.a	What are dominators? Draw a flow graph with at least one loop and obtain the dominators of each node.	4

Answer b or c

b For the flow graph shown below perform live variable analysis and explain 8



c Write the algorithm for finding the available expression and explain with the help of a suitable example **8**

Q.no.	Module 6	Marks
6.a	Explain true dependence and name dependence with the help of suitable examples.	4

Answer b or c

b Explain global code motion in detail **8**

c Explain the concept of loop unrolling with the help of suitable examples. **8**