Total No. of Questions: 6 *Total No. of Printed Pages:3*

Enrollment No.....

Faculty of Science

End Sem Examination May-2024

BC3CO55 Compiler Design

Programme: B.Sc. Branch/Specialisation: Computer Science

Duration: 3 Hrs. **Maximum Marks: 60**

Note: All questions are compulsory. Internal choices, if any, are indicated. Answers of Q.1 (MCQs) should be written in full instead of only a, b, c or d. Assume suitable data if necessary. Notations and symbols have their usual meaning.

Q.1 i. How many tokens in a given program code? **1**

```
main ( ) {
  a=b + + + - - - - + + + = = ;
  printf ( “sum %d%d” , a ,b ) ;
}
```

(a) 30 (b) 26 (c) 25 (d) 24

ii. Consider the following C program: **1**

```
int main ( )
{
  int i , n ;
fro ( i=0 ; i ≤ n ; i++)
}
```

What is the compiler response about the program?

(a) Compiler produces lexical error

(b) Compiler produces syntax error

(c) Compiler produces lexical & syntax error

(d) None of these

iii. Consider the following grammar then which of the following are the handle detect to parse the string w=n+n*n **1**

(a) E→n / E+n / E+n*n

(b) E→n / E+n / E+E*n

(c) E→n / E+n / n+n*n

(d) E→n / E+n / E*n

P.T.O.

iv. Grammar G are { S→FR, R→*S / ε, F→id }. Choose the correct option for M[S , id] & M[R , $] {Here $ is dollar symbol & ε is null symbol} **1**

(a) {S→FR} & { R→ε } (b) {S→FR} & { R→*S }

(c) {F→id} & { R→ε } (d) {F→id} & { }

v. What is not true about data flow analysis? **1**

(a) Useful in register allocation

(b) Dead code elimination is not possible

(c) Eliminates common sub expression

(d) Used in constant & variable propagation

vi. Which of the following is a machine independent optimization? **1**

(a) Constant folding (d) Copy propagation

(c) Peephole optimization (d) Loop jamming

vii. Why is intermediate code Generation based on an abstract machine model useful in compilers? **1**

(a) Implementation of lexical analysis and syntax analysis is made easier

(b) Portability of the front end of the compiler

(c) Writing for intermediate code generation

(d) All of these

viii. Consider the basic block given below **1**

{ a=b+c, c=a+d, d=b+c, e=d-b, a=e+b }

The minimum number of nodes & edges present in the DAG representation of the above basic block respectively are…..

(a) 8 & 10 (b) 9 & 12 (c) 4 & 4 (d) 6 & 6

ix. YACC is a computer program for _____ operation system. **1**

(a) Open SUSE (b) Unix

(c) Window (d) DOS

x. Which of these is not true about the Symbol Table? **1**

(a) All the labels of the instructions are symbols

(b) Table has entry for symbol name address value

(c) Perform the processing of the assembler directives

(d) Created during pass 1

Q.2 i. What is a translator with a block diagram? **2**

ii. Describe the input buffering with different buffering schemes. **3**

iii. Explain the phases of the compiler with an example. **5**

OR iv. Explain the different stages of translation & execution of a program. **5**

Q.3 i. What is three address codes (TAC)? Explain with an example. **2**

ii. Calculate the first and follow functions for the given grammar & also construct a predictive parsing table. **8**

S → ACB / CbB / Ba

A → da / BC

B → g / ∈

C → h / ∈

OR iii. Design LALR(1) parsing table for the given grammar. **8**

{ E→E + T / T / TF , T→F & F→F * / a / b }

Q.4 i. Define basic block, flow graph & leader. **3**

ii. Explain Global data flow analysis with example. **7**

OR iii. What is code optimization? Explain different types of code optimization techniques. **7**

Q.5 i. Give an example to show how DAG is used for register allocation. **4**

ii. What is code generation? Explain different properties of code generation. **6**

OR iii. Explain run time storage management with any example. **6**

Q.6 Attempt any two:

i. Explain symbol table in detail. **5**

ii. What is grammar? Explain different types of grammar. **5**

iii. Describe LEX & YACC. **5**

******

Marking Scheme

BC3CO55 (T) Compiler Design

Q.1	i)	C	**1**
	ii)	B	**1**
	iii)	D	**1**
	iv)	A	**1**
	v)	B	**1**
	vi)	C	**1**
	vii)	A	**1**
	viii)	D	**1**
	ix)	B	**1**
	x)	C	**1**
Q.2	i.	Definition of translator-1 Mark & block diagram-1 Mark	**2**
	ii.	Define input buffering-1 Mark & schemes-2 Marks	**3**
	iii.	Compiler phases-3 Marks & example with passing different phases-2 Marks	**5**
OR	iv.	Stages diagram-2 Marks & explain different components-3 Marks	**5**
Q.3	i.	TAC define-1 Mark & Example-1 Mark	**2**
	ii.	Calculated first value with proper solution steps-3 Marks & Calculated follow value with proper solution steps-3 Marks & Predictive parsing table-2 Marks	**8**
OR	iii.	Complete LALR(1) solution steps-6 Marks & LALR(1) Parsing table-2 Marks	**8**
Q.4	i.	Each define-1 Mark	**3**
	ii.	Explain Global data flow analysis-3 & numerical Example-4	**7**
OR	iii.	Definition of code optimization-2 Marks & Optimization techniques-5 Marks (Each 1 Mark & give maximum 5 marks)	**7**
Q.5	i.	Explain examples in details-4 Marks	**4**
	ii.	Define code generation-2 Marks & Explain each property 1 Mark(maximum 4 marks)	**6**
OR	iii.	Explain run time storage management with diagram-4 Marks & example-2 marks	**6**
Q.6			
	i.	Details explanation of symbol table-5 Marks	**5**
	ii.	Define grammar-1 Mark & explain different types of grammars-4 Marks	**5**
	iii.	LEX-2.5 Marks & YACC-2.5 Marks	**5**

******