

PES UNIVERSITY, Bangalore

UE18CS351

End Semester Examination - B. Tech Sem 6 - July, 2021 Department of Computer Science & Engineering (Established under Karnataka Act No. 16 of 2013) **UE18CS351 - COMPILER DESIGN**

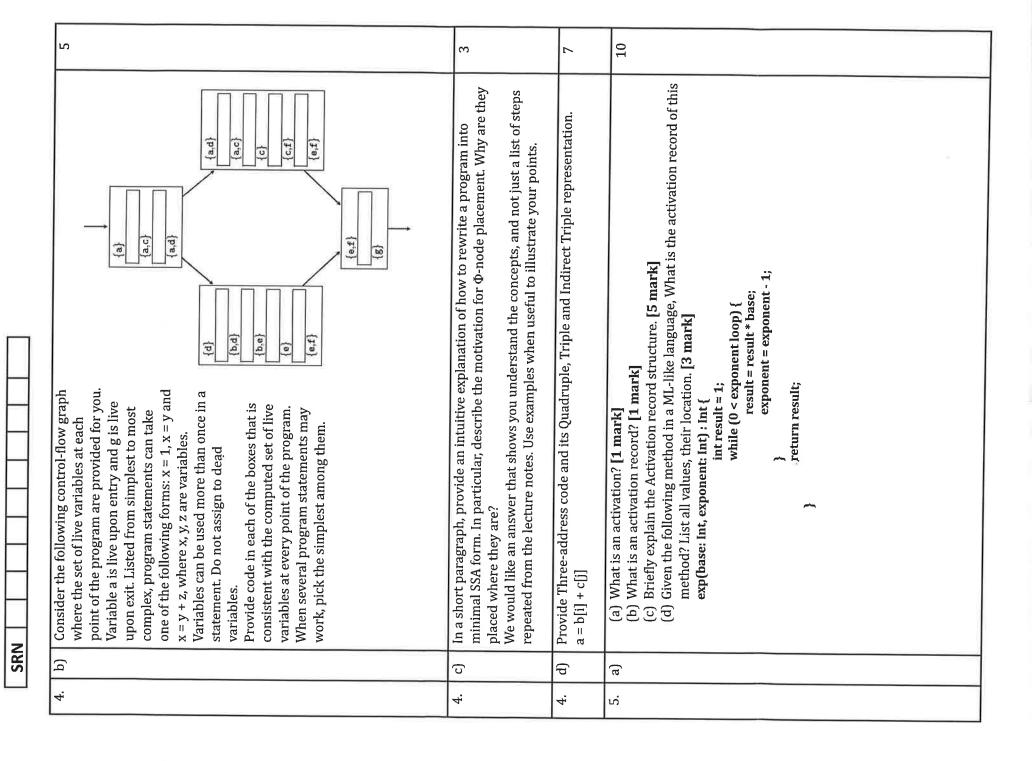
Time: 3 hrs.

Answer All Questions

Max Marks: 100

10 10 2 3 Consider the following tokens and their associated regular expressions, given as a flex scanner What is a Transition Diagram? Draw Transition Diagram for an Identifier. How can you handle (a) Give an input to this scanner such that the output string is bccaa. [Mention clearly the The following is a grammar for an abstracted Lisp-like language, in which every expression is represented as a list containing an operator and any number of operands, which can in turn (b) What will be the output for the input string: 0110010011 (Provide tokenized input) (a) Construct the DFA for viable prefixes of this grammar using LR(0) items. [5 marks](b) Identify any shift-reduce and reduce-reduce conflicts in this grammar under SLR(1) With an example and a diagram explain the various phases of a compiler. You must clearly (c) Is this grammar in LL(1)? **[2 marks]** (d) Can any of the bottom up parsers you studied in the course, parse the grammar? (Try answering the questions without constructing any parse table) (a) What is Follow(E), Follow(B), Follow(U), Follow(V)? [4 marks] part of the input that matches each of the letter in the string (b) Is the grammar left recursive? [2 marks] With an example explain the following terms: depict every input and output to each phase. Given the grammar answer the following: keywords separately from identifiers? printf (" b ") printf (" a ") (1010*1|0101*0) printf("c") (c) Viable Prefix [1 mark] $E \to UV \mid EBE \mid V \mid [E]$ be expressions or identifiers. (b) Reduction [1 mark] (a) Handle [1 mark] rules. [2 marks] E o (op C | id [2 marks] $C \rightarrow) \mid E C$ (ob C $B \rightarrow ? | ! | @$ specification: $V \to a \mid b$ $\stackrel{\wedge}{\longrightarrow} n$ 0(01)*1(01|10)(q a C a **p** \overline{C} ij ij 2 3 2

SRN



SRN