## Question 1

0 / 0 pts

This is a dummy question to allow you to upload scanned or typed answers for the Lexical Analyzer question below.

Please upload a single file for your answer.

☁ **Exam iitk.pdf**

> extra page of some other course !!!

## Question 2

10 / 10 pts

Consider a C program, starting at line 1 in a file:

```
int main() { /* my program */
    int x, y;
    If ( x < y)
        return TRUE;
}
```

Assume a standard lexical analyzer for C language. Complete the following table for ALL the tokens generated by the lexical analyzer in proper sequence. If there is any lexical error, enter **ERROR** in the token field, and a short reason for the error in the lexeme field. Continue to fill the rest of the table from the next valid token/lexeme.

 You can choose your own token names, but they should be informative.

-----------------------------------------------------------------

token            lexeme            line#

-----------------------------------------------------------------

Your Answer:

## Question 3

0 / 0 pts

**Unanswered**

This is a dummy question to allow you to upload scanned or typed answers for the CFG question below.

Please upload a single file for your answer.

## Question 4

21 / 35 pts

The set of regular expression $R$ over the alphabet $\{a, b\}$ can be defined as follows:

1. $\epsilon$ (the empty string) is a regular expression ($\epsilon \in R$)
2. $a$ and $b$ are regular expressions ($a \in R, b \in R$)
3. if $r \in R$ and $s \in R$ then
   1. $r + s \in R$
   2. $r \cdot s \in R$
   3. $r* \in R$

Now answer the following questions.

a) Write a context free grammar (G) that accepts the set of regular expressions as its language. (5 marks)

b) Is your grammar (G) ambiguous? Justify your answer. (4 marks)

c) If needed, augment the grammar G. Construct the FIRST and FOLLOW set for ALL non-terminals present in the augmented grammar G. (6 marks)

d) If G is ambiguous, rewrite it to make it unambiguous. You can assume the precedence of various operators present in G as: $*$ (highest), $\cdot$ (second), $+$ (third). All the operators are LEFT associative. You can ignore this part if your grammar is already unambiguous. (8 marks. To be added to part (a) if G is already unambiguous)

e) Construct the SLR parse table for the unambiguous form of G (for 12 marks). If you cannot, you can choose to construct the parse table for ambiguous G itself (for 7 marks).

**Suggestion**: The symbol · can cause confusion since it is used in the language as well as in the construction of LR(...) items. Feel free to use another symbol in one of the places to avoid confusion.

Your Answer: