INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Date .....FN / AN *Time:* 2/3 Hrs. *Full Marks* 45 *No. of Students* 91
Autumn / ~~Spring~~ Semester, 2011-2012 Deptt. CSE Sub No. CS31003
3RD Yr. B. Tech.(Hons.) / ~~B-Arch~~. / ~~M. Sc~~. Sub. Name COMPILERS

Instructions : **Answer All The Questions.**

1. Translate the following C function to GCC-x86 assembly language code. Write brief comments in each line explaining the translation. **[10]**

```
int bad(int n){
    int i, sum=0;
    for(i=0; i<=n; ++i) sum += i;
    return sum;
}
```

Assume that addresses of `n`, `i` and `sum` are `(ebp) + 8`, `(ebp) − 4`, and `(ebp) − 8` respectively. Following are a few x86 assembly language instructions:
`addl cmpl jle jmp leave movl pushl ret subl`

2. The object language alphabet Σ is the set of all ASCII characters except the following 13 characters: { · { } [ ] ( ) + ∗ ? \ | " }. They have special meaning in the expressions. These characters are included in Σ using \ e.g \[is treated as '['. **[5 + 5 + 5]**

   (a) Give a *context-free grammar*(CFG) for the set of regular expressions defined as follows:
      i. ε, $\emptyset$, $a \in \Sigma$, $\backslash n$, $\backslash t$ are regular expressions.
      ii. . (dot), "x", [x], where x $\in \Sigma^+$ are regular expressions.
      iii. If r and s are regular expressions, then so are (r), $(r|s)$, (rs), $r*$, $r+$, $r?$, $r\{m, n\}$ are regular expressions, where $0 \leq m \leq n$

   (b) Give a regular expression for the floating-point numbers written either in e (2.5e+1) or in f (25.0) format. Sign of the number is not part of it.

   (c) Give a DFA, corresponding to the regular expression of part (b) (no formal construction is required).

3. Consider the following ambiguous grammar with the usual meaning of the operators:

$$
\begin{aligned}
E &\rightarrow B \mid A \\
B &\rightarrow B \text{ or } B \mid B \text{ and } B \mid \text{not } B \mid (B) \mid \textit{true} \mid \textit{false} \mid R \\
R &\rightarrow A < A \mid A = A \mid A <= A \\
A &\rightarrow A + A \mid A - A \mid A * A \mid A / A \mid (A) \mid -A \mid *A \mid \texttt{ic} \mid \texttt{id}
\end{aligned}
$$

   The non-terminals are $\{E, B, R, A\}$ with E as the start symbol. **[5 + 5 + 10]**

   (a) Write an equivalent unambiguous grammar for the language so that the natural precedence and associativity rules are embedded in the grammar. Orders of precedence are:
   or < and < not; {+ −} < {∗ /} < {− ∗} (unary).

   (b) Remove *left-recursions* from the unambiguous grammar you have got in (a), and *left-factor* the production rules if necessary. Is the transformed grammar $LL(1)$?

   (c) Write pseudo code for a *recursive-descent predictive* parser for the Boolean expression part of the grammar starting from the non-terminal B. Assume A to be a terminal.