Analysis part is often called _____ of the compiler where as the synthesis is the _____ of the compiler

a. Front end, Back end
b. Checking, Correcting
c. View, Phase
d. Crux, Core

Ans: a

Which of the following is not an application of Compiler Technology?

a. Type Checking
b. Binary Translation
c. Implementation of High-Level Programming languages
d. Firmware Implementation

Ans: d

Missing ';' in a 'C' language programming statement is an example for

a. Lexical Error
b. Syntax Error
c. Semantic Error
d. Logical Error

Ans: b

Semantic Consistency refers to _____

a. Error free Program
b. Type checking
c. Automatic error correction
d. Automatic error detection

Ans: b

Which of the following statement is an example for Lexical error in a 'C' program?

a. int float double register;
b. a = b + c;
c. _a = b_2 + 3_c;
d. a_ *= b_1;

Ans: c

Deleting successive characters from the input until a well-formed token is found is an example of _____ error recovery strategy.

a. Panic mode
b. Phrase level recovery
c. Error Productions
d. Global Corrections

Ans: a

Consider the Grammar below

W -> w(C)S | w(C){S}

C -> rLC | r | ε

L -> && | II | ε

S -> a;S | ε

Is the grammar LL (1)?

a. Yes
b. Yes only after Left factoring productions from non terminals 'W' and 'S'
c. Yes only after removing left recursion from 'W' and 'C'
d. Yes only after Left factoring productions from non terminals 'W' and 'C'

Ans: d

Backtracking in Recursive descent parsing refers to _____

a. Changing terminals on stack to match the user input string
b. Applying the productions for non terminal to match the user input string
c. Changing the production already applied for the non terminal to match the input string
d. None of the above

Ans: c

Predictive Parser requires

a. Backtracking
b. Computing First & Follow
c. LL (1) Grammar
d. Both b & c

Ans: d

Which of the following is the rule for Left Recursion Elimination where α, β and ɣ are strings?

a. A ->αA | ɣ replaced with A-> ɣA' and A' -> αA' | ε
b. A -> Aβ | ɣ replaced with A - > βA' and A' -> ɣA' | ε
c. A ->A ɣ | α replaced with A -> αA' and A' -> ɣA' | ε
d. A -> βA | α replaced with A -> βA' and A' -> αA' | ε

Ans: c

Grammar of the programming is checked at _____ phase of compiler.

(A) Syntax analysis

(B) Semantic analysis

(C) Code generation

(D) Code optimization

Ans: A

Identify the cause for representing the syntax by a grammar

(A) It is concise

(B) It is accurate

(C) Automation becomes easy

(D) All of the above

Ans: D

CFG (Context Free Grammar) can be recognized by a

(A) Push down automata

(B) Finite state automata

(C) 2 way linear bounded automata

(D) Both a and c

Ans: D

Semantic errors can be detected at

(A) Compile time only

(B) Run-time only

(C) Both (a) and (b)

(D) None of these

Ans: C

Left factoring is the process of factoring out the common

(A) Prefixes of alternates

(B) Suffixes of alternates

(C) Both(a) and (b)

(D) None of these

Ans: A

The cost of developing a compiler is proportional to

(A) Flexibility of the available instruction set

(B) Complexity of the architecture of the target machine

(C) Complexity of the source language

(D) All of these

Ans: D

Which among the following is/are interpreted language?

(A) C++

(B) Java

(C) Visual basic

(D) Both B and C

Ans: D

The number of tokens in the following C statement is

```
if(a>b)
 printf("RVCE ISE");
```

(A) 4

(B) 8

(C) 13

(D) 15

Ans: C

Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?

(A) Removing left recursion alone

(B) Factoring the grammar alone

(C) Removing left recursion and factoring the grammar

(D) None of these

Ans: D

Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order

(A) Leftmost Derivation

(B) Leftmost derivation traced out in reverse

(C) Rightmost Derivation

(D) Rightmost Derivation traced out in reverse

Ans: A

_____ is process of finding a parse tree for a string of tokens

(A) Analysing
(B) Recognizing
(C) Tokenizing
(D) Parsing
Ans: D

The data structure which allows us to find the record for each identifier quickly and to store or retrieve data from that record quickly is _____

(A) Operator Table
(B) Symbol Table
(C) Keyword Table
(D) Definition Table
Ans: B

A grammar G is said to be ambiguous if _____
(A) Generates more than one parse tree for the same string
(B) Leftmost and Rightmost derivations are same for the given sentence
(C) Grammar has more than one ε production
(D) Only (A) and (B)
Ans: D

Identify the issue in Top down Parsing
(A) Left recursion
(B) Ambiguity
(C) Backtracking
(D) All of the above
Ans: D

Consider the following lex code

%{

```
#include<stdio.h>
int  p=0,s=0,m=1;
FILE *fp1,*fp2;
%}
%option noyywrap
%%
"scanf" {s++;fprintf(yyout,"readf");}
"printf" {p++;fprintf(yyout,"writef");}
. {fprintf(yyout,yytext);}
%%
```

Code can be used for

(A) Counting number of times printf and scanf statements appear in a 'C' program
(B) Replacing printf and scanf with writef and readf in a 'C' program
(C) Both A & C
(D) None of the above

Consider the lex program segment

```
%{
#include <stdio.h>
int chr=0,sp=0,lin=0,wds=0,m=1;
%}
%option noyywrap
%%
[^ \t\n]+ {ds++;hr+=yyleng;}
[\n] {nl++;hr++;}
[ ] {ps++;hr++;}
%%
```

The above lex code can be used for

(A) Counting number of lines, characters, words and spaces
(B) Counting number of tab space and newline
(C) Counting hours, seconds and minutes elapsed
(D) Only A & B

Ans: A

Variables yylval and yyleng has _____ and _____ in any Lex program

(A) value associated with token, length of matched string
(B) pointer to matched string, value associated with token
(C) pointer to matched string, length of matched string
(D) token, matched string

Ans: A

Lex executes the action for _____ for the current input

(A) smallest possible match
(B) longest possible match
(C) Both A & B
(D) None of the above

Pattern [0-9]*\.[0-9]+ does not match

(A) 0.0
(B) 4.5
(C) 0.31415
(D) 2

Ans: D

Which of the following is not a part of Lex program sections

(A) subroutine
(B) rules
(C) definitions
(D) actions