Read the questions carefully and check exactly ONE option. Write your answer to the answer sheet

Question 1: *Which of the following grammars is not LL(1)?*

A. S → 1SA, S → 0A1, S → 2, A → 0A1, A → 1
B. S → aAS | b, A → cS | d, B → c
C. S → aSa | bSb | cSc | d
D. S → ε | ab | ba | aSb | bSa

Question 2: *Consider the context free grammar:*
A →B x | y C | ε, B → C z A, C →x B, where {A,B,C} is the set of nonterminal symbols,{x, y, z} is the set of terminal symbols A is the start symbol. What is FIRST (A)?

A. {y, ε} C. {y,z, ε}
B. {x,y, ε} D. {x, y, z, ε}

Question 3: *A top down parser generates*

A. Sequence of productions used in rightmost derivation
B. Sequence of productions used in rightmost derivation in reverse
C. Sequence of productions used in left most derivation
D. Sequence of productions used in left most derivation in reverse

Question 4: *...........__ is considered as an instance of a token.*

A. Texeme
B. Pattern
C. Lexeme
D. Mexeme

Question 5*: The lexical analyzer takes__...........___as input and produces a stream of ....as output.*

A. Source program, tokens
B. Token, source program
C. Grammar, Source program
D. Regular expression, tokens

Question 6:*When is the type checking is usually done?*

A. During code optimizaton
B. During lexical analysis
C. During syntax directed translation
D. During syntax analysis

Question 7: *A parse tree showing the value of attributes at each node is*

A. Annotated Parse Tree
B. Attribute Parse Tree
C. Semantic Tree
D. Syntax Tree

Question 8*: Which of the following is a top down parser?*

A. SLR parser
B. LALR parser
C. Operator precedence parser
D. Recursive descent parser.

Question 9: *The main difference between a sentence and a sentential form is*

A. there is no difference;
B. a sentence contains only terminal symbols but a sentential form can contain some non-terminal symbols
C. sententialforms are a subset of sentences but the converse is not true
D. sentences are derived from S but sentential forms are not

Question 10: In which situation, inhereted attribute is natural choice*:*

A. Evaluation of arithmetic expressions
B. Keeping track of variable declarations.
C. Checking for correct use of L-value and R - value
D. None of the above.

Question 11: *Task of the lexical analysis is*

A. To parse the source program into the basic elements or tokens of the language
B. To build a literal table and an identifier table
C. To build a uniform symbol table
D. All of these

Question 12: *Right parse is*

A. The sequence of productions used in an arbitrary derivation of a from S.
B. Reversion of the sequence of productions used in rightt derivation of a from S
C. The sequence of productions used in right derivation of a from S
D. None of the above

Question 13: *Program counter in stack calculator is used to?*

A. store the top of stack address
B. store the base address of stack
C. to store the middle address of the stack
D. to store the next instruction address

Question 14: *Consider the grammar shown below*

S → i E t S S' | a
S' → e S | ε
E → b

In the predictive parse table. M, of this grammar, the entries M[S', e] and M[S', $] respectively are

A. {S' → e S} and {S' → ε}

phai tim follow(S')

B. {S' → e S} and Error

C. {S' → e} and {S' → ε}

D. {S' → e S, S'→ ε} and {S' → ε}

Question 15: *Backus-Naur Form (BNF) is a notation for which of the following:*

A. context-free grammars

B. context-sensitive grammars

C. unrestricted grammars

D. all of the above

Question 16: *The least number of temporary variables required to create a three-address code in static single assignment form for the expression q + r/3 + s – t * 5 + u * v/w is*

A. 4

B. 8

khi nao them lenh moi: chi cho nao co toan tu thi them thuoc tinh place => dem so toan tu = 8

C. 7

D. 9

Question 17: *Left recursion is not permitted for top down parsing and right recursion is not permited for bottom up parsing*

A. Yes B. No

Question 18: *Under which of the following circumstances might you choose to implement a programming language using a compiler rather than an interpreter?*

A. Executables for programs in the language should be able to be distributed and executed without the language implementation.

B. Programs in the language need to perform well (run quickly

C. The language allows the program to generate and execute program code in the language dynamically

D. You would like programmers to be able to detect program flaws statically

Question 19*: Which of the following instructions is written in three address code?*

A. t[i]:=x[i+j]+1

B. t[i]:=1

C. t[i]:=1+x[i]

D. a:=- t[i]

dieu kien cho ma 3 dia chi:
+ dung toi da 1 toan tu
(lay gia tri index trong array cung la 1 toan tu) => chi co B

Question20 : *Given grammar S→aSb, S →c and string aacbb. Which of the following is the next configuration of (q, 2, S**1**aS**1**,aSbb#)?*

A. (q, 3, S1aS1aS2, cbb#)

B. (q, 3, S1aS1a, Sbb#)

C. (b, 2, S1aS1,aSbb#)

D. None of the above

Question 21: *Reduction in strength in code optimization means*

A.replacing run-time computation by compile time computation

B.replacing a costly operation by a relatively cheaper one

C.Both (a) & (b)

D.removing loop invariant computation

Question 22: *Consider the syntax directed definition shown below.*

S → id : = E {gen (id.place = E.place;);}
E → E1 + E2 {t = newtemp ( ); gen (t = El.place + E2.place;); E.place = t}
E → id {E.place = id.place;}

Here, gen is a function that generates the output code, and newtemp is a function that returns the name of a new temporary variable on every call. Assume that ti's are the temporary variable names generated by newtemp. For the statement 'X: = Y + Z', the 3-address code sequence generated by this definition is

A. X = Y + Z

B. t1 = Y + Z; X = t1

C. t1 =Y; t2 = t1 + Z; X = t2

D. t1 = Y; t2 = Z; t3 = t1 + t2; X = t3

Question 23: *Grammar* $E \to TE', E' \to +TE' \mid \varepsilon, T \to FT', T' \to *FT' \mid \varepsilon, F \to id \mid (E)$ *is*

A. Ambiguous

B. Depends on given string

C. ambigouos for certain pair of terminaaaaaaaal

D. Unambiguous

Question 24: *The graph that shows basic blocks and their successor relationship is called*

A. Directed Acyclic Graph

B. Control Flow Graph

C. Flowchart

D. Syntax graph

Question 25: *Which of the following optimizations can be applied to the following code*

```
(1)   prod := 0
(2)   i := 1
(3)   t1 := 4 * i
(4)   t2 := a[t1]
(5)   t3 := 4 * i
(6)   t4 := b[t3]
(7)   t5 := t2 * t4
(8)   t6 := prod + t5
(9)   prod := t6
(10)   t7 := i + 1
(11)   i := t7
(12)   if i <= 20 goto (3)
(13)   …
```

A. Dead code elimination   cac bien chi su dung 1 lan
B. Common Sub-expression Elimination   neu co vp giong nhau
C. Constant Propagation   co thang nao do tinh dua vao bieu thuc trc
D. Partial redundancy elimination

Question 26. *Which is not a code optimization strategy?*

A. Constant folding
B. Copy propagation
C. Dead code elimination
D. Control flow graph

Question 27: *How many tokens are there in the following assignment* $\overset{1}{\underline{ac}}$ $\overset{2}{\underline{:=}}$ $\overset{1}{\underline{ba}}$ *(*1.) of KPL?*   2

A. 6
B. 7
C. 8
D. None of the above

sau (* thi ko tinh

Question 28: *What is the value of X printed by the following KPL program ?*

```
program COMPUTE;
var X : integer ;
procedure FIND ( X: integer) ;
             begin
                  X : = X*X ;
                                  end ;
              begin
                            X : = 2
                          FIND ( X );
                          call writeI ( X );
             end
```

tham tri (pass by value), chi chay o trong chinh ham find

A. 2
B. 4
C. 8
D. 16

Question 29: *Recursive descent parser is an example of*

A. Top down backtracking parser
B. Bottom up backtracking parser
C. Predictive parser
D. None of the above

Question 30: *The following object code*

L1:
  <code of condition>
  FJ L2
  <code of statement>
  J L1
L2:
is generated from which statement?

A. if <condition> then <statement>   if thi thuong co thuoc tinh true va false
B. if <condition> then <statement> else <statement>
C. while <condition> do<statement>
D. do <statement> while <condition>

Question 31*: Consider the context free grammar {L → TL', L' → ∨TL', L' → ε, T → PT', T' → ∧PT', T' → ε, P → i, P → (L)}.What is Follow(T')?*

A. {(, ∨}   C. {∨, $,)}
B. {∧,), ε}   D. None of the above

Follow(T) = First(L') v Follow(L) = {v,), e}
Follow(T') = Follow(T)....

Question 32: *How many blocks are there in the following code*

1 X := 4
2 Y := 5
3 X := X + Y
4 Z := 10

```
5 A :=Y*Z;
6 B := X / 2 ;
7 if B < Z goto 12
8 P := Z - 1
9 Z := Z - P
10 C := Y
11 goto 7
12 R :=A * Y
13 if X < 10 goto 4
14. P := C
```

A. 2 B.3 C. 4 D. 6

Question 33: *A grammar that produces zero or one parse tree for some sentence is called*

a) Ambiguous

b) Unambiguous

c) Regular

d) None of these

Question 34: *LR stands for*

a) Left to right

b) Left to right reduction

c) Right to left

d) Left to right and right most derivation in reverse

1/ Write a program in KPL to do the following tasks:

- Read n integers (n ≤ 10) into array A
- Print the input values from last to first.
- Print the sum of the elements that are evenly divisable by 5.

2/ Prove that the following syntax rules:

84) <Term2> ::= SB_TIMES <Factor> <Term2>

85) <Term2> ::= SB_SLASH <Factor> <Term2>

86) <Term2> ::= ε

satisfy LL(1) condition.

FIRST(RHS84) = {SB_TIMES}
FIRST(RHS85) = {SB_SLASH}
FIRST(RHS86) = {e}
FIRST(RHSi) ^ FIRST(RHSj) = empty, for all i =/= j

FOLLOW(Term2) = {SB_PLUS, SB_MINUS, ...} (trong slide)
=> The set does not include SB_TIMES and SB_SLASH, so
+ FIRST(RHS84) ^ FOLLOW(Term2) = empty
+ FIRST(RHS85) ^ FOLLOW(Term2) = empty
=> Conclusion: Production of <Term2> satisfies LL(1).

** dieu kien cua LL(1):
+ FIRST1(ai) ^ FIRST1(aj) = empty, for all i =/= j
+ FIRST1(ai) ^ FOLLOW1(A) = empty, for all i =/= j