

1000-compilers-questions-answers

1. Number of states of FSM required to simulate behaviour of a computer with a memory capable of storing “m” words, each of length ‘n’.

- a) $m \times 2^n$
- b) 2^{mn}
- c) $2^{(m+n)}$
- d) all of the mentioned

Answer: b

Explanation: For every Data here length is n and memory's state is defined in terms of power of 2, Here the total memory capability for all the words = mn Hence the number of states is 2^{mn} .

2. An FSM with _____

- a) M can be transformed to Numeral relabeling its states
- b) M can be transformed to N, merely relabeling its edges
- c) Both of the mentioned
- d) None of the mentioned

Answer: c

Explanation: The Definition of FSM states that M can be transformed to N by relabeling its states or its edges.

3. Which of the following statement is correct?

- a) A Context free language can be accepted by a deterministic PDA
- b) union of 2 CFLs is context free
- c) The intersection of two CFLs is context free
- d) The complement of CFLs is context free

Answer: b

Explanation: Context-free languages are closed under the following operations. The Kleene star, the concatenation, the union and the intersection.

Which of the following is true?

- a) Only S1 is correct
- b) Only S2 is correct
- c) Both S1 and S2 are correct
- d) None of S1 and S2 is correct

Answer: c

*Explanation: S1 can be written as $(00)^n$ where $n \geq 1$. And S2 can be written as $(00)^{(m+n)}$ where $m \geq 2$ and $n \geq 1$. S2 can be further reduced to $(00)^x$ where $x \geq 3$. SO we can write regular grammars for both
G1 -> G100/00 (For S1)
G2 -> G200/000000 (For S2).*

5. Which of the following pairs of regular expressions are equivalent?

- a) $1(01)^*$ and $(10)^*1$
- b) $x(xx)^*$ and $(xx)^*x$
- c) x^+ and $x^+x^{(*)+}$
- d) All of the mentioned

Answer: d

*Explanation: Rule $(pq)^*p=p(qp)^*$*

Therefore— $(xx^)(x^*x^{**})$*

$(xx^)(x^*x^*)$ [Using $x^{**}=x$] $(xx^*)(x^*)$ [Using $x^*x^*=x^*$] (xx^*) [Using $x^*xx^*=x^*$]*

x^+

6. Given a NFA with N states, the maximum number of states in an equivalent minimized DFA is at least.

- a) N^2
- b) 2^N
- c) $2N$
- d) $N!$

Answer: b

Explanation: The initial state of the DFA constructed from this NFA is the set of all NFA states that are reachable from state 1 by ϵ -moves; that is, it is the set $\{1, 2, \text{ and } 3\}$. A transition from states 1, 2, and 3 by input symbol 0 must follow either the arrow from state 1 to 2, or from state 3 to 4. Also, neither state 2 nor 4 have outgoing ϵ -moves.

7. Let L denotes the language generated by the grammar S – OSO/00. Which of the following is true?

- a) $L = O$
- b) L is regular but not O
- c) L is context free but not regular
- d) L is not context free

Answer: b

Explanation: The grammar itself is not regular but language L is regular as L can be represented using a regular grammar, for example $S \rightarrow S00/00$.

8. Which of the following are not regular?

- a) String of)'s which has length that is a perfect square
- b) Palindromes Consisting of 0's 1's
- c) String of 0's whose length is a prime number
- d) All of the mentioned

Answer: d

Explanation: Strings of odd number of zeroes can be generated by the regular expression $(00)^*0$. Pumping lemma can be used to prove the non-regularity of the other options.

9. If $\Sigma = \{a, b, c, d, e, f\}$ then number of strings in Σ of length 4 such that no symbol is used more than once in a string is

- a) 35
- b) 360
- c) 49
- d) 720

Answer: b

Explanation: Here string length is 4 so we create string of length 4 by 6 values firstly we arrange any value by 6 methods. Then Remaining numbers are 5 so we can arrange them by 5 methods then remaining numbers are 4 so we arrange them by 4 methods and then 3. Thus $6*5*4*3=360$.

10. Which one of the following statement is FALSE?

- a) Context-free languages are closed under union
- b) Context-free languages are closed under concatenation
- c) Context-free languages are closed under intersection
- d) Context-free languages are closed under Kleene closure

Answer: c

Explanation: CFL is closed under Kleene closure, concatenation, and Union

- a) aabb
- b) abab
- c) aababb
- d) aaabbb

Answer: d

Explanation: Given $S \rightarrow SaSbS | \epsilon$

$S \rightarrow aSbS$ putting $S \rightarrow \epsilon$ and then $S \rightarrow SaSbS$

$S \rightarrow aSaSaSbSbSbS$ putting $S \rightarrow SaSbS$

$S \rightarrow aaabbb$ putting $S \rightarrow \epsilon$. Answer: b

Explanation: RE is used for all types of string and numbers.

3. What is the Regular Expression Matching Zero or More Specific Characters?

- a) x
- b) #
- c) *
- d) &

Answer: c

*Explanation: Zero or Specific Expression matching can be done only by a single character that is *.*

4. All _____ are automatically treated as regular expressions.

- a) Programmatic description
- b) Window
- c) Win Object
- d) Collection

Answer: a

Explanation: It is seen that programmatic description are treated as regular expression. Answer: a

Explanation: XML checkpoints employ RE.

6. The production Grammar is $\{S \rightarrow aSbb, S \rightarrow abb\}$ is _____ grammar:

- a) Type-3
- b) Type-2
- c) Type-1
- d) Type-0

Answer: b

Explanation: As per the definition of type-2 grammar.

7. Regular expression $(x/y)(x/y)$ denotes which of the following set?

- a) $\{xy, xy\}$
- b) $\{xx, xy, yx, yy\}$
- c) $\{x, y\}$
- d) $\{x, y, xy\}$

Answer: b

Explanation: From first part if we take x then from the latter part x then it forms xx

From first part if we take x then from the latter part y then it forms xy

From first part if we take y then from the latter part x then it forms yx

From first part if we take y then from the latter part y then it forms yy.

8. Regular expression x/y denotes which of the following set?

- a) $\{x, y\}$
- b) $\{xy\}$
- c) $\{x\}$
- d) $\{y\}$

Answer: a

Explanation: Because either x or y can be selected.

9. The regular expressions denote zero or more instances of an x or y is?

- a) $(x+y)$
- b) $(x+y)^*$

c) $(x^* + y)$

d) $(xy)^*$

Answer: b

Explanation: For instances of x or y the exp is $x+y$ and both can zero or more times than $(x+y)^*$.

1. A system program that combines separately compiled modules of a program into a form suitable for execution is called _____

- a) Assembler
- b) Linking loader
- c) Cross compiler
- d) None of the mentioned

Answer: b

Explanation: The definition of cross compiler.

2. A compiler for a high-level language that runs on one machine and produces code for a different machine is called _____

- a) Optimizing compiler
- b) One pass compiler
- c) Cross compiler
- d) Multipass compiler

Answer: c

Explanation: A compiler for a high-level language that runs on one machine and produces code for a different machine is called cross compiler.

3. Cross-compiler is a compiler _____

- a) Which is written in a different language from the source language?
- b) That generates object code for the machine it's running on.
- c) Which is written in the same language as the source language?
- d) That runs on one machine but produces object code for another machine

Answer: a

Explanation: A compiler for a high-level language that runs on one machine and produces code for a different machine is called a cross compiler.

Answer: a

Explanation: Bootstrapping to a new platform. When software is developed for a new platform, a cross compiler is used to compile necessary tools such as the OS and a native compiler.

Answer: a

Explanation: GCC, a free software collection of compilers, also can be used as cross compile. It supports many languages and platforms.

6. The _____ is a technique for building cross compilers for other machines.

- a) Brazilian Cross
- b) Canadian Cross
- c) Mexican Cross
- d) X-cross

Answer: b

Explanation: The Canadian Cross is a technique for building cross compilers for other machines. Given three machines X , Y , and Z , one uses machine X (e.g. running Windows XP on an IA-32 processor) to build a cross compiler that runs on machine Y (e.g. running Mac OS X on an x86-64 processor) to create executables for machine Z .

7. _____ was developed from the beginning as a cross compiler.

- a) Free Pascal
- b) GCC
- c) Pascal
- d) None of the mentioned

Answer: a

Explanation: Free Pascal was developed from the beginning as a cross compiler. The compiler executable (ppcXXX where XXX is target architecture) is capable of producing executables for all OS of the same architecture.

1. If we compile the sam.c file with the command “gcc -o sam sam.c”, then the executable file will be?

- a) a.out**
- b) sam**
- c) sam.out**
- d) None of the mentioned**

Answer: b

Explanation: This is how the GCC is designed to take names of executable files.

2. What will be output of the following code?

```
#include<stdio.h>
int main()
{
    printf("%d\t", sizeof(6.5));
    printf("%d\t", sizeof(90000));
    printf("%d", sizeof('A'));
    return 0;
}
```

- a) 8 4 2**
- b) 8 4 2**
- c) 8 4 4**
- d) 8 4 3**

Answer: c

Explanation: GCC compilers (32 bit compilers) size of:

double is 8 byte

long int is 8 byte

Character constant is 2 byte.

3. What will be output of the following c code? (according to GCC compiler)

```
#include<stdio.h>
int main()
{
    signed x;
    unsigned y;
    x = 10 +- 10u + 10u +- 10;
    y = x;
    if(x==y)
        printf("%d %d", x, y);
    else if(x!=y)
        printf("%u %u", x, y);
    return 0;
}
```

- a) 0 0**
- b) 65536 -10**
- c) 0 65536**
- d) Compilation error**

Answer: a

Explanation: Consider on the expression:

$x = 10 +- 10u + 10u +- 10;$

10: It is signed integer constant.

10u: It is unsigned integer constant.

X: It is signed integer variable.

As we know operators enjoy higher precedence than binary operators. So

$x = 10 + (-10u) + 10u + (-10);$

$= 10 + -10 + 10 + (-10);$

$= 0$

So, Corresponding signed value of unsigned 10u is +10.

4. What will be output of the following c code?

```
#include<stdio.h>
int main()
{
    const int *p;
    int a=10;
    p=&a;
    printf("%d",*p);
    return 0;
}
```

a) 0

b) 10

c) Garbage Value

d) Any Memory address

Answer: b

Explanation: In the following declaration

*const int *p;*

p can keep address of constant integer.

5. What will be output of the following c code?

```
#include<stdio.h>
int main()
{
    int a= sizeof(signed) +sizeof(unsigned);
    int b=sizeof(const)+sizeof(volatile);
    printf("%d",a+++b);
    return 0;
}
```

a) 10

b) 9

c) 8

d) Error

Answer: c

Explanation: Default data type of signed, unsigned, const and volatile is intSo, a = 4 and b =4

Now, a+++b

$= a++ + b$

$= 4 + 4$ //due to post increment operator.

$= 8$

But in Linux gcc compiler size of int is 4 byte so your out will be 16.

6. Which of the following is integral data type?

a) void

b) char

c) float

d) double

Answer: b

Expanation: In c char is integral data type. It stores the ASCII value.

7. What will be output of the following c code?

```
#include<stdio.h>
int main()
{
    volatile int a=11;
    printf("%d",a);
    return 0;
}
```

- a) 11
- b) Garbage
- c) -2
- d) Cannot Predict

Answer: d
Explanation: Value of volatile variable can't be predicted because its value can be changed by any microprocessor interrupt.

8. What will be output of the following c code?

```
#include<stdio.h>
const enum Alpha
{
    X,
    Y=5,
    Z
}p=10;
int main()
{
    enum Alpha a,b;
    a= X;
    b= Z;
    printf("%d",a+b-p);
    return 0;
}
```

- a) -4
- b) -5
- c) 10
- d) 11

Answer: a
Explanation: Default value X is zero and
 $Z = Y + 1 = 5 + 1 = 6$
So, $a + b - p$
 $= 0 + 6 - 10 = -4$.

1. What is the output of lexical analyzer?

- a) A set of RE
- b) Syntax Tree
- c) Set of Tokens
- d) String Character

Answer: c
Explanation: A lexical analyzer converts character sequences to set of tokens.

2. Which symbol table implementation is based on the property of locality of reference?

- a) Linear list
- b) Search tree
- c) Hash Table
- d) Self Organisation

Answer: c

Explanation: Hash table is used as a reference for symbol table because it is efficient.

3. Which of the following is true for operator precedence parsing?

- a) For all pair of non-terminal**
- b) For all pair of non-terminals**
- c) To delimit the handle**
- d) None of the mentioned**

Answer: a

Explanation: There are two important properties for these operator precedence parsers is that it does not appear on the right side of any production and no production has two adjacent non-terminals. Implying that no production right side is empty or has two adjacent non-terminals. So accordingly to property option (A) is correct.

4. What is an Object program?

- a) Program written in machine language**
- b) Program to be translated into machine language**
- c) Translation of high-level language into machine language**
- d) None of the mentioned**

Answer: c

Explanation: Since the input is the source language and the output that we get after the analysis is the machine language.

5. Which concept of FSA is used in the compiler?

- a) Lexical analysis**
- b) Parser**
- c) Code generation**
- d) Code optimization**

Answer: a

Explanation: Because the lexer performs its analysis by going from one stage to another.

6. Which concept of grammar is used in the compiler?

- a) Lexical analysis**
- b) Parser**
- c) Code generation**
- d) Code optimization**

Answer: b

Explanation: As the lexical analysis of a grammar takes place in phases hence it is synonymous to parser.

7. Which of the following are Lexemes?

- a) Identifiers**
- b) Constants**
- c) Keywords**
- d) All of the mentioned**

Answer: d

Explanation: Different Lexical Classes or Tokens or Lexemes Identifiers, Constants, Keywords, Operators.

1. What constitutes the stages of the compilation process?

- a) Feasibility study, system, design, and testing**
- b) Implementation and documentation**
- c) Lexical analysis, syntax, Analysis and code generation**
- d) None of the mentioned**

Answer: c

Explanation: As defined in the compilation process.

- 2. The lexical analyzer takes _____ as input and produces a stream of _____ as output.**
- a) Source program, tokens**
 - b) Token, source program**
 - c) Either of the two**
 - d) None of the mentioned**

Answer: a

Explanation: As per the definition of Lexical Analyser which states that lexical analysis is the process of converting a sequence of characters into tokens.

- 3. Parsing is also known as _____**
- a) Lexical Analysis**
 - b) Syntax Analysis**
 - c) Semantic Analysis**
 - d) Code Generation**

Answer: b

Explanation: Parsing or syntactic analysis is the process of analysing a string of symbols and conforming to the rules of grammar.

- 4. A compiler program written in a high level language is called _____**
- a) Source Program**
 - b) Object Program**
 - c) Machine Language Program**
 - d) None of the mentioned**

Answer: a

Explanation: The input that we give in high level language is also known as the source language.

- 5. System program such a compiler are designed so that they are _____**
- a) Re-enterable**
 - b) Non-Usable**
 - c) Serially usable**
 - d) None of the mentioned**

Answer: a

Explanation: For the convince of the user compilers are made re-enterable.

- 6. Which of the following is not a feature of compiler?**
- a) Scan the entire program first and translate into machine code**
 - b) To remove syntax errors**
 - c) Slow for debugging**
 - d) Execution time is more**

Answer: d

Explanation: The objective of the compiler is clearly not to increase the execution time of the program.

- 7. A system program that brings together separately compiled modules of a program into a form language that is suitable for execution.**
- a) Assembler**
 - b) Linking loader**
 - c) Cross compiler**
 - d) None of the mentioned**

Answer: b

Explanation: A loader which brings together the functions of a relocating loader with the ability to combine a number of

program segments that have been independently compiled into an executable program.

8. A programmer by mistakes writes a program to multiply two numbers instead of dividing them, how can this error be detected?

- a) Compiler
- b) Interpreter
- c) Compiler or interpreter
- d) None of the mentioned

Answer: d

Explanation: This is a logical error that can't be detected by any compiler or interpreter. Answer: a

Explanation: Because it checks for all the values and determines whether the output string matches with the given string. Answer: b

Explanation: No not for all values the string and numbers can we use the RE. Answer: b

Explanation: For every cycle the values does not change unpredictably because the type of grammar that it accepts is defined.

4. All _____ are automatically treated as regular expressions.

- a) Programmatic description
- b) Window
- c) Win Object
- d) Collection

Answer: a

Explanation: The programmatic description is genuinely treated as regular expression. Answer: a

Explanation: The backslash carries no significance and it is ignored.

6. The regular expression denotes a language comprising all possible strings of even length over the alphabet (0, 1).

- a) $1 + 0(1+0)^*$
- b) $(0+1)(1+0)^*$
- c) $(1+0)$
- d) $(00+0111+10)^*$

Answer: d

Explanation: Option $1 + 0(1+0)^* \rightarrow$ It does not consider even length criteria for the question.

Option $(0+1)(1+0)^* \rightarrow$ It can so happen here that from the former bracket it takes 0 or 1 and takes null from the latter then it forms a string of odd length

Option $(1+0) \rightarrow$ It gives either 1 or 0.

Hence Option $(00+0111+10)^*$ is the answer.

7. The RE gives none or many instances of an x or y is?

- a) $(x+y)$
- b) $(x+y)^*$
- c) $(x^* + y)$
- d) $(xy)^*$

Answer: b

Explanation: Whether x or y is denoted by $x+y$ and for zero or more instances it is denoted but $(x+y)^*$.

1. The RE in which any number of 0's is followed by any number of 1's followed by any number of 2's is?

- a) $(0+1+2)^*$
- b) $0^*1^*2^*$
- c) $0^* + 1 + 2$
- d) $(0+1)^*2^*$

Answer: b

Explanation: The order for the desired string is 012 and for any number of 0s we write 0^* for any number of 1s we denote it by 1^* and similarly for 2^* . Thus $0^*1^*2^*$.

2. The regular expression have all strings of 0's and 1's with no two consecutive 0's is?

- a) $(0+1)$**
- b) $(0+1)^*$**
- c) $(0+\epsilon)(1+10)^*$**
- d) $(0+1)^*011$**

Answer: c

Explanation: From the former bracket we choose 0 or epsilon. Then from the latter part 1 or 10 which can be followed by 1 or 10.

3. The regular expression with all strings of 0's and 1's with at least two consecutive 0's is?

- a) $1 + (10)^*$**
- b) $(0+1)^*00(0+1)^*$**
- c) $(0+1)^*011$**
- d) $0^*1^*2^*$**

Answer: b

*Explanation: The expression $(0+1)^*00(0+1)^*$ is where either it initially takes 0 or 1 or 00 followed by string of combination of 0 and 1.*

4. Which of the following is NOT the set of regular expression $R = (ab + abb)^*bbab$?

- a) ababbbbab**
- b) abbbab**
- c) ababbabbbab**
- d) abababab**

Answer: d

Explanation: abababab doesn't end with bbab whereas the other 3 options satisfy the given regular expression.

- a) aabccd**
- b) adabcca**
- c) abcca**
- d) abababd**

Answer: a

Explanation: $S \rightarrow aS$ (substitute $S \rightarrow aS$)

$S \rightarrow aaS$ (substitute $S \rightarrow bA$)

$S \rightarrow aabA$ (substitute $A \rightarrow ccA$)

$S \rightarrow aabccA$ (substitute $A \rightarrow d$)

$S \rightarrow aabccd$.

6. Consider the production of the grammar $S \rightarrow AA$ $A \rightarrow aa$ $A \rightarrow bb$ Describe the language specified by the production grammar:

- a) $L = \{aaaa, aabb, bbba, bbbb\}$**
- b) $L = \{abab, abaa, aaab, baaa\}$**
- c) $L = \{aaab, baba, bbba, bbbb\}$**
- d) $L = \{aaaa, abab, bbba, aaab\}$**

Answer: a

Explanation: $S \rightarrow AA$ (substitute $A \rightarrow aa$)

$S \rightarrow aaaa$

$S \rightarrow AA$ (substitute $A \rightarrow aa$)

$S \rightarrow aaA$ (substitute $A \rightarrow bb$)

$S \rightarrow aabb$

$S \rightarrow AA$ (substitute $A \rightarrow bb$ the $A \rightarrow aa$)

$S \rightarrow bbba$

$S \rightarrow AA$ (substitute $A \rightarrow bb$)

S->bbbb.

7. If R is regular language and Q is any language (regular/ non regular), then Pref (Q in R) is _____

- a) Non-regular
- b) Equal
- c) Infinite
- d) Regular

Answer: d

Explanation: So says the definition of Regular Grammar. Answer: b

Explanation: Here the non terminal that gives null will said to have a null production.

1. (a,b) what is a?

- a) Domain
- b) Range
- c) Domain & Range
- d) None of the mentioned

Answer: a

Explanation: A is called the domain.

2. (a,b) what is b?

- a) Domain
- b) Range
- c) Domain & Range
- d) None of the mentioned

Answer: b

Explanation: B is called the Range. Answer: a

Explanation: All the elements of A are related with itself by relation R, hence it is a reflexive relation. Answer: a

Explanation: a is related to b by R, and if b is also related to a by the same relation R. Answer: a

Explanation: a is related to b by R, and b is related to c by R, and similarly for a and c.

6. The smallest set A such that $A \cup \{1, 2\} = \{1, 2, 3, 5, 9\}$ is?

- a) {2,3,5}
- b) {1, 2, 5, 9}
- c) {3, 5, 9}
- d) None of the mentioned

Answer: c

Explanation: Given $A \cup \{1, 2\} = \{1, 2, 3, 5, 9\}$. Hence $A = \{3, 5, 9\}$.

7. If a set A has n elements, then the total number of subsets of A is?

- a) N
- b) 2^n
- c) N^2
- d) $2n$

Answer: b

Explanation: Number of subsets of A = ${}^nC_0 + {}^nC_1 + \dots + {}^nC_n = 2^n$.

1. If $A \cap B = B$, then?

- a) $A \subset B$
- b) $A = \emptyset$
- c) $B \subset A$
- d) $B = \emptyset$

Answer: c

Explanation: Since $A \cap B = B$, hence $B \subset A$.

2. Empty set is a _____

- a) Invalid set
- b) Infinite set
- c) Finite set
- d) None of the mentioned

Answer: c

Explanation: Empty set is a finite set.

3. If A, B and C are any three sets, then $A - (B \cup C)$ is equal to _____

- a) $(A - B) \cup (A - C)$
- b) $(A - B) \cup C$
- c) $(A - B) \cap (A - C)$
- d) $(A - B) \cap C$

Answer: c

Explanation: it is De' Morgan law.

4. $A = \{x: x \neq x\}$ represents?

- a) $\{0\}$
- b) $\{1\}$
- c) $\{\}$
- d) $\{x\}$

Answer: c

Explanation: That is a fact.

5. If A, B, C be three sets such that $A \cup B = A \cup C$ and $A \cap B = A \cap C$, then?

- a) $A=B$
- b) $A=C$
- c) $B=C$
- d) $A=B=C$

Answer: c

Explanation: Transition Law.

6. The number of proper subsets of the set $\{1, 2, \text{ and } 3\}$ is?

- a) 8
- b) 6
- c) 7
- d) 5

Answer: b

Explanation: Number of proper subsets of the set $\{1, 2, 3\} = 2^3 - 2 = 6$.

7. If A and B are any two sets, then $A \cup (A \cap B)$ is equal to _____

- a) A
- b) B
- c) $A \cap B$
- d) $A \cup B$

Answer: a

Explanation: $A \cap B \subseteq A$ Hence $A \cup (A \cap B) = A$.

8. If A, B and C are any three sets, then $A \times (B \cup C)$ is equal to _____

- a) $(A \times B) \cup (A \times C)$
- b) $(A \times B) \cap (A \times C)$
- c) $(A \cup B) \times (A \cup C)$
- d) None of the mentioned

Answer: a

Explanation: It is distributive law.

1. A language L from a grammar $G = \{ VN, \Sigma, P, S \}$ is?

- a) Set of symbols over VN
- b) Set of symbols over Σ
- c) Set of symbols over P
- d) Set of symbols over S

Answer: b

Explanation: The definition of the grammar is set of symbols over Σ .

2. What is the transitional function of a DFA?

- a) $Q \times \Sigma \rightarrow Q$
- b) $Q \times \Sigma \rightarrow 2Q$
- c) $Q \times \Sigma \rightarrow 2n$
- d) $Q \times \Sigma \rightarrow Qn$

Answer: a

Explanation: Q is the finite set and let be a finite set of symbols so $Q \times \Sigma$ gives no of states.

3. What is the transitional function of an NFA?

- a) $Q \times \Sigma \rightarrow Q$
- b) $Q \times \Sigma \rightarrow 2Q$
- c) $Q \times \Sigma \rightarrow 2n$
- d) $Q \times \Sigma \rightarrow Qn$

Answer: b

Explanation: Let Q be a finite set and let be a finite set of symbols. Also let be a function from Q to $2Q$. All the elements of Q a state, the transition function, q_0 the initial state and A the set of accepting states. Then a nondeterministic finite automaton is a 5-tuple $\langle Q, \Sigma, q_0, \delta, A \rangle$.

4. Maximum number of states of a DFA converted from an NFA with n states is?

- a) n
- b) n^2
- c) 2n
- d) None of the mentioned

Answer: c

Explanation: Take the NFA with states $\{q_0, q_1\}$, alphabet $\Sigma = \{a\}$, initial state q_0 , transitions $\delta(q_0, a) = q_0$, $\delta(q_0, a) = q_1$ and final state q_1 . It generates the same language as the DFA with the same set of states and alphabet, but transitions $\delta(q_0, a) = q_1$ and $\delta(q_1, a) = q_1$.

5. What are the basic limitations of finite state machine?

- a) It cannot remember arbitrarily large amount of information
- b) It cannot remember state transitions
- c) It cannot remember grammar for a language
- d) It cannot remember language generated from a grammar

Answer: b

Explanation: Because it does not store its previous state of the transition.

6. The string WWR is not recognized by any FSM because _____

- a) An FSM cannot remember arbitrarily large amount of information
- b) An FSM cannot fix the midpoint
- c) An FSM cannot match W with WR
- d) An FSM cannot remember first and last inputs

Answer: b

Explanation: Palindromes cannot be recognized by FSM.

7. A finite automata recognizes _____

- a) Any Language
- b) Context Sensitive Language
- c) Context Free Language
- d) Regular Language

Answer: d

Explanation: All regular languages are implemented by finite automata.

1. Which of the following statement is true for Dead State?

- a) It cannot be reached anytime
- b) There is no necessity of the state
- c) If control enters no way to come out from the state
- d) If control enters FA deads

Answer: c

Explanation: It is a rejecting state for if the control enters it reaches the dead end and cannot reach an accepting state.

2. Which of the following statement is true for Moore Machine?

- a) Output depends on present state
- b) Output depends on present input
- c) Output depends on present state and present input
- d) Output depends on present state and past input

Answer: a

Explanation: The definition states that moore machines output is determined by the current state only.

3. Which of the following statement is true for Mealy Machine?

- a) Output depends on present state
- b) Output depends on present input
- c) Output depends on present state and present input
- d) Output depends on present state and past input

Answer: c

Explanation: The definition states that its output is determined by current state and current input.

4. Which is true for in accessible state?

- a) It cannot be reached anytime
- b) There is no necessity of the state
- c) If control enters no way to come out from the state
- d) If control enters FA deads

Answer: a

Explanation: The very meaning of in accessible state is that it cannot be reached at any point of time.

5. In Mealy Machine O/P is associated with _____

- a) Present state
- b) Next state
- c) Input
- d) None of the mentioned

Answer: b

Explanation: The definition states that its output is determined by current state and current input.

6. In Moore Machine O/P is associated with _____

- a) Present state**
- b) Next state**
- c) Input**
- d) None of the mentioned**

Answer: a

Explanation: The definition states that moore machines output is determined by the current state only.

7. Which type of string is accepted by the following finite automata?

- a) All string**
- b) Null string**
- c) No string**
- d) None of the mentioned**

Answer: b

Explanation: Null strings are not accepted by finite automata.

8. Myhill-Nerode Theorem is used for _____

- a) Minimization of DFA**
- b) Maximization of NFA**
- c) Conversion of NFA**
- d) Conversion of DFA**

Answer: a

Explanation: Myhill–Nerode theorem provides a necessary and sufficient condition for a language to be regular. The Myhill–Nerode theorem can be generalized to trees. And used for minimization of DFA.

a)

b)

c)

d) None of the mentioned

Answer: a

Explanation: We can verify that the string ababa is accepted by this NFA once we “guess” the state path $q_0, q_2, q_5, q_2, q_5, q_2 \in F$. Of course the only choice is the first one. If we made the wrong start q_0, q_1, q_3, q_4, q_1 we reach a point where we have a remaining a to process with no place to go. This is a failure.

- a)
- b)
- c)
- d) None of the mentioned

Answer: b
Explanation: A ϵ transition takes no input and represents a pure nondeterministic choice of being in the state or the target state without having done any processing.

3. NDFAs were introduced by _____
- a) Michael O Rabin & Dana Scott
 - b) Dan Brown
 - c) Sun micro system Labs
 - d) SAP Labs

Answer: a
Explanation: NFAs were introduced Dana Scott and Michael O. Rabin who also showed their equivalence to DFAs.

4. The regular languages are not closed under _____
- a) Concatenation
 - b) Union
 - c) Kleene star
 - d) Complement

Answer: d
Explanation: RE are closed under

- Union (cf. picture)
- Intersection
- Concatenation
- Negation
- Kleene closure.

5. The Tuples for NDFA is _____
- a) $\Sigma, Q, q_0, F, \delta$
 - b) Q, q_0, F, δ
 - c) $\Theta, Q, q_0, F, \delta$
 - d) $F, Q, \Delta, q_0, \delta$

Answer: a
Explanation: An NFA is represented formally by a 5-tuple, $(Q, \Sigma, \Delta, q_0, F)$, of

- a set of states Q
- a set of input symbols Σ
- a transition function $\Delta : Q \times \Sigma \rightarrow P(Q)$.
- an initial state $q_0 \in Q$
- a final state $F \subseteq Q$.

6. NFAs are _____ DFAs.

- a) Larger than**
- b) More expressive than**
- c) Less expressive than**
- d) Equally expressive as**

Answer: a

Explanation: Because there is more number of states for an NFA than for a DFA for a given expression.

7. An NFA's transition function returns _____

- a) A Boolean value**
- b) A state**
- c) A set of states**
- d) An edge**

Answer: c

Explanation: A transition function $\Delta: Q \times \Sigma \rightarrow P(Q)$. Where $P(Q)$ denotes the power set of Q .

1. Conversion of a DFA to an NFA _____

- a) Is impossible**
- b) Requires the subset construction**
- c) Is Chancy**
- d) Is nondeterministic**

Answer: b

Explanation: In order to convert NFA to DFA we work with sets of state where each state in the DFA corresponds to a set of NFA states.

2. A regular language corresponds to _____

- a) An alphabet**
- b) Set of strings over an alphabet**
- c) A DFA only**
- d) A DFA or an NFA**

Answer: b

Explanation: A regular grammar takes in all strings over an alphabet.

3. An NFA may be converted to a DFA using _____

- a) Induction**
- b) A construction**
- c) Contradiction**
- d) Compilation**

Answer: b

Explanation: subset construction is used to convert a NFA into DFA.

4. The subset construction shows that every NFA accepts a _____

- a) String**
- b) Function**
- c) Regular language**
- d) Context-free language**

Answer: c

Explanation: Like DFAs, NFAs only recognize regular languages.

a)

b)

c)

d) None of the mentioned

Answer: a

Explanation: The NDFA initially takes either a or b followed by a then b then reaches the final state or takes iterations of a or b to reach the final state.

6. Which is the application of NFA?

a) A regular language is produced by union of two regular languages

b) The concatenation of two regular languages is regular

c) The Kleene closure of a regular language is regular

d) All of the mentioned

Answer: d

Explanation: As per its definition.

Answer: a
Explanation: Therefore it is possible to convert an existing NFA into a DFA for the purpose of implementing a simpler machine. Which is executed by using the powerset construction.

Answer: a
Explanation: It is a known fact.

1. Can a DFA simulate NDFA?

a) No

b) Yes

c) Sometimes

d) Depends on NDFA

Answer: b

Explanation: Yes it can be done through power set construction.

2. Find the wrong statement?

a) The language accepted by finite automata are the languages denoted by regular expression

b) Every DFA has a regular expression denoting its language

c) For a regular expression r , there does not exist NDFA with $L(r)$ and transit that accept

d) None of the mentioned

Answer: c

Explanation: The vice versa is true.

3. Regular expression a/b denotes which of the following set?

- a) {a}
- b) {ε,a,b}
- c) {a,b}
- d) {ab}

Answer: c

Explanation: Either a is the output or b hence it's {a, b}.

4. Which behaviour of a NFA can be stimulated by DFA?

- a) Always
- b) Sometimes
- c) Never
- d) Depends on NFA

Answer: a

Explanation: It can be done through power set construction.

Answer: a
Explanation: It the standard procedure to convert NFA to DFA.

6. What is the relation between NFA-accepted languages and DFA accepted languages?

- a) >
- b) <
- c) =
- d) <=

Answer: c

Explanation: The no of languages accepted by NFA and DFA is equal.

7. In regular expressions, the operator ‘*’ stands for?

- a) Concatenation
- b) Selection
- c) Iteration
- d) Addition

Answer: c

Explanation: It indicates iterations which can vary from zero to any number.

Answer: a
Explanation: Two options are invalid because they both accept ‘b’ as a string which is not accepted by give DFA. D is invalid because it accepts bb^+a which are not accepted by given DFA.

2. The lexical analysis for a modern language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?

- a) Finite state automata
- b) Deterministic pushdown automata
- c) Non-deterministic pushdown automata
- d) Turing machine

Answer: a

Explanation: Initially in lexical analysis the program is divided into tokens. Tokens can be expressed as regular expressions: $[a-zA-Z][a-zA-Z0-9]^*$

the keyword if is given by if.

Integers are given by $[+]?[0-9]^+$.

Answer: a
Explanation: Empty strings can be inputted n a NDFA.

Answer: b
Explanation: The given alphabet Σ contains only one symbol {a} and the given NFA accepts all strings with any number of occurrences of ‘a’. Hence the complement is an empty string.

- a) i, ii and iii
- b) ii, iii and iv
- c) i, ii and iv

d) i, iii and iv

Answer: c

Explanation: Any combination of strings in set {ab, aa, baa} will be in L^ .*

i) "abaabaaabaa" can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are "ab aa baa ab aa"

ii) "aaaabaaaa" can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are "aa ab aa aa"

iii) "baaaaabaaaab" cannot be partitioned as a combination of strings in set {ab, aa, baa}

iv) "baaaaabaa" can be partitioned as a combination of strings in set {ab, aa, baa}. The partitions are "baa aa ab aa".
Answer: d

Explanation: State 'q' is trap state. All other states are accepting states. In state 00, DFA must move to 'q' for input symbol 0. All (non-trap) states indicate names indicate the characters seen before reaching that particular state.

a) i, ii, iii, iv

b) i, ii

c) ii, iii, iv

d) iii, iv

Answer: d

Explanation: i) Is a variation of Turing Machine Halting problem and it is undecidable.

....ii) Context Free Languages are not closed under intersection and complement.

....iii) Complement of Regular languages is also regular.

....iv) Recursive Languages is closed under complement.

8. The definition of a language L with alphabet $\{a\}$ is given as following. $L = \{ ank \mid k > 0, \text{ and } n \text{ is a positive integer constant} \}$ What is the minimum number of states needed in a DFA to recognize L ?

a) $k+1$

b) $n+1$

c) $2n+1$

d) $2k+1$

Answer: b

Explanation: Note that n is a constant and k is any positive integer. For example, let $n=3$, then the DFA must be able to accept $aaa, aaaaaa, aaaaaaaaa, \dots$ build such a DFA, 4 states are required.

1. $S \rightarrow aSa \mid bSb \mid a \mid b$; the language generated by the above grammar is the set of _____

a) All palindromes

b) All odd length palindromes

c) Strings beginning and ending with the same symbol

d) All even length palindromes

Answer: b

Explanation: The strings accepted by language are $\{a, b, aaa, bbb, aba, bab, \dots\}$. All the strings are odd length palindromes.

a) strings with the substring 00

b) strings with at most two 0's

c) strings with at least two 0's

d) strings beginning and ending with either 0 or 1

Answer: c

Explanation: The RE having 2 0's padded by $(0+1)^$ which means accepted strings must have at least 2 0's.*

3. Which one is a FALSE statement?

a) There exists a unique DFA for every regular language

- b) NFA can always be converted to a PDA
- c) Complement of CFL is always recursive
- d) Every NDFA can be converted to a DFA

Answer: d

Explanation: Deterministic PDA cannot handle languages or grammars with ambiguity, but NDFA can handle with ambiguous languages as well as context-free grammar. Hence not every Ndfa can be converted to DFA.

4. Match the following.

Group 1	Group 2
P. Regular expression	1. Syntax analysis
Q. Pushdown automata	2. Code generation
R. Dataflow analysis	3. Lexical analysis
S. Register allocation	4. Code optimization

- a) P-4, Q-1, R-2, S-3
- b) P-3, Q-1, R-4, S-2
- c) P-3, Q-4, R-1, S-2
- d) P-2, Q-1, R-4, S-3

Answer: b

Explanation: Regular grammar relates to lexical analysis

Pushdown automata relates to Syntax analysis

Data flow analysis is Code optimization

Register allocation is code generation.

- Then L is?
- a) Not recursive
 - b) Regular
 - c) Context free but not regular
 - d) None of the mentioned

Answer: c

Explanation: The language L1 accept strings {c, abc, abcab, aabbcab, aabbcaabb,} and L2 accept strings {a, b, c, ab, abc, aabc, aabbc, ... }. Intersection of these two languages is $L1 \cap L2 = \{akbkc \mid k \geq 0\}$ which is not regular but context free.

Answer: a

Explanation: Initially Q is empty. Then since the initial state of the DFA is {0}, {0} is added to Q.

Since $2(0, a) = \{1, 2\}$, {1, 2} is added to Q and $(\{0\}, a) = \{1, 2\}$.

Since $2(0, b) = ,$ is added to Q and $(\{0\}, b) = .$

At this point $Q = \{\{0\}, \{1, 2\}, \}$. Similarly $(\{1, 2\}, b) = \{1, 3\}$. Hence {1, 3} is added to Q. Similarly $(\{1, 3\}, a) = \{1, 2\}$ and $(\{1, 3\}, b) = .$ Thus there are no new states to be added to Q. Since the transitions from all states of Q have been computed and no more states are added to Q, the conversion process stops here.

Answer: b

Explanation: It adds nothing new to the automata.

Answer: a

Explanation: NFA-ε can be transformed into a NFA always, the properties are also true for NFAs.

Answer: a

Explanation: The ε-closure of a set of states Z of an NFA is defined as the set of states reachable from any state in Z following ε-transitions.

Answer: a

Explanation: ε-transitions provides a convenient transition in the systems whose current states are not precisely known.

4. Which of the following CFG’s can’t be simulated by an FSM?
- a) S->Sa/b
 - b) S->aSb/ab
 - c) S->abX, X->cY, Y->d/aX
 - d) None of the mentioned

Answer: b

Explanation: generates the set {an bn, n=1,2,3}which is not regular.

5. The transitions which does not take an input symbol are called _____

- a) ϵ -transitions
- b) λ -transitions
- c) ϵ -transitions & λ -transitions
- d) none of the mentioned

Answer: c

Explanation: The transitions taking an input symbol are called ϵ -transitions or λ -transitions. Answer: a

Explanation: Both are equivalent. Answer: a

Explanation: Yes ordinary NFA and NFA- ϵ are the same, in that, given either one, one can construct the other, which recognizes the same language.

8. Which of the following is a correct statement?
- a) { If an $bn \mid n = 0,1, 2, 3 \dots$ } is regular language
 - b) Strings with equal number of a's and b's denies a regular language
 - c) $L(A^* B^*) \cap B$ gives the set A
 - d) None of the mentioned

Answer: c

Explanation: If we include A and B in a set and if we write A^* it means except then A i.e. B same as B^* means except then B i.e. so if we intersect $(A^* B^*)$ and B then get A because in any regular language. If we write $A-B$ then $A-B=A$ intersection B' so if we intersect A and B means $A-B$ So the intersection of $(A^* B^*)$ and $B = (BA)$. intersection B means $(BA)-B'$ and $B'=A$ so (BA) intersection $(A)=A$. Answer: c

Explanation: The states traversed are ABDBDABDAC, and the only edge not traversed C D. Answer: a

Explanation: 111001 cannot be the suffix of any string accepted by this DFA. Suppose $s=w111001$. No matter what state the DFA reaches after reading w, it will go to state D after reading "111", then go to state B after reading "00" and finally reaches state C after reading "1".

3. Find the pair of regular expressions that are equivalent.
- a) $(0+1)^*$ and $(0^*+1^*)^*$
 - b) $(0+1)^*$ and $(0+1^*)^*$
 - c) $(0+10)^*$ and $(0^*+10)^*$
 - d) All of the mentioned

Answer: d

Explanation: All generate all strings of 0's and 1's thus are these pairs are equivalent.

4. Which of the following strings is NOT in the Kleene star of the language {011, 10, 110}?
- a) 01
 - b) 10
 - c) 110
 - d) 10011101

Answer: d

Explanation: Every string in the language $\{011, 10, 110\}^*$ has to be formed from zero or more uses of the strings 011, 10, and 110. A string may be used more than once.

5. Which grammar is not regular?
- a) 0^n
 - b) $0^n 1^n n$
 - c) $0^m 0^n n$
 - d) $0^n 0^n n$

Answer: a

Explanation: According to pumping lemma, is not a regular language. It is the language of the DFA with two states to achieve an even number of 0's...

6. If L is a language, and a is a symbol, then, the quotient of L and a , is the set of strings such that wa is in L . Suppose L is regular, which of the following statements is true?

- a) L/a is always a regular language
- b) L/a is not a regular language
- c) All of the mentioned
- d) None of the mentioned

Answer: a

Explanation: We can build a DFA for as such: firstly we get the DFA for: Then, we copy all the states and transitions to the DFA for. However, we mark any state as a final state in if and only if is a final state in.

7. Here is a context-free grammar $G: S \rightarrow AB \mid A \rightarrow 0A1 \mid 2 \mid B \rightarrow 1B \mid 3A$ which of the following strings are in $L(G)$?
- a) 021300211
 - b) 022111300211
 - c) None of the mentioned
 - d) 021300211 & 022111300211

Answer: d

Explanation: First, notice that A generates strings of the form $0^n 2 1^n$, where n is 0 or more. Also, B gives zero or more 1's, which is followed by one 3, and then A gives something. Since S generates something an A can generate followed by something a B can generate, the strings in $L(G)$ are of the form $0^n 2 1^m 3 0^n 2 1^n$. Answer: b

Explanation: $S \Rightarrow AB \Rightarrow AbA \Rightarrow Aba \Rightarrow aSba \Rightarrow aABba \Rightarrow aAbAba \Rightarrow aAbaba \Rightarrow aababa$.

9. The grammar $G: S \rightarrow SS \mid a \mid b$ is ambiguous. Check all and only the strings that have exactly two leftmost derivations in G .
- a) bbb
 - b) ab
 - c) All of the mentioned
 - d) None of the mentioned

Answer: c

Explanation: $S \Rightarrow a$. A string of length 2 has only one derivation, e.g., $S \Rightarrow SS \Rightarrow aS \Rightarrow ab$.

10. For the following grammar: $S \rightarrow A \mid B \mid 2 \mid A \rightarrow C0 \mid D \mid B \rightarrow C1 \mid E \mid C \rightarrow D \mid E \mid 3 \mid D \rightarrow E0 \mid S \mid E \rightarrow D1 \mid S$ Identify all the unit pairs.
- a) D,C
 - b) A,B
 - c) B,C
 - d) A,C

Answer: a

Explanation: The cycle of unit-productions $S \rightarrow A \rightarrow D \rightarrow S$ says that any pair involving only S , A , and D is a unit pair. Similarly, the cycle $S \rightarrow B \rightarrow E \rightarrow S$ tells us that any pair involving S , B , and E is a unit pair.

- a) P is not closed under union
- b) NP is not closed under intersection
- c) None of the mentioned
- d) P is not closed under union & NP is not closed under intersection

Answer: d

Explanation: Both P and NP are closed under each of these operations. Answer: a

Explanation: They both are equivalent.

3. _____ a part of a compiler that is responsible for recognizing syntax.
- a) Parser
 - b) Bzr
 - c) Linker
 - d) Optimizer

Answer: a

Explanation: Parser recognises all the syntax of the language.

4. _____ a part of a compiler that takes as input a stream of characters and produces as output a stream of words along with their associated syntactic categories.

- a) Parser**
- b) Optimizer**
- c) Scanner**
- d) None of the mentioned**

Answer: c

Explanation: A compiler's scanner reads an input stream that consists of characters and produces an output stream that contains words, each labelled with its Syntactic category.

5. _____ an IR-to-IR transformer that tries to improve the IR program in some way.

- a) Optimizer**
- b) Parser**
- c) All of the mentioned**
- d) None of the mentioned**

Answer: a

Explanation: The optimizer is an IR-to-IR transformer that tries to improve the IR program in some way.

6. _____ a phase of a compiler that maps the IR program into the instruction set and the finite resources of the target machine.

- a) Optimizer**
- b) Parser**
- c) Optimizer & Parser**
- d) None of the mentioned**

Answer: d

Explanation: In a two-phase compiler, ensures that there is a source program and an object program. Answer: a

Explanation: nfa has n then dfa has at max 2^n . Answer: a

Explanation: An NFA is nothing more than an ϵ -NFA with no ϵ transitions. Thus $\delta(q, \epsilon)$ for all states $q = \emptyset$. Answer: a

Explanation: For every DFA, there is an ϵ -NFA that accepts the same language and Vice Versa. Answer: a

Explanation: For every NFA there is an ϵ -NFA that accepts a similar language and vice versa.

1. Consider the languages $L_1 = \{a\}$ and $L_2 = \{a\}$. Which one of the following represents $L_1 L_2^* \cup L_1^*$?

- a) ϵ**
- b) a^***
- c) All of the mentioned**
- d) None of the mentioned**

Answer: a

Explanation: $L_1^ = \{a\}^*$ which is $\{a\}^*$. Answer: c*

Explanation: The DFA can be minimized to two states and the second state is final state. We reach second state after a 0.

- a) 1, 2, 3**
- b) 2, 3, 4**
- c) 1, 2, 4**
- d) 1, 3, 4**

Answer: c

Explanation: The given alphabet Σ contains only one symbol $\{a\}$ and the given NFA accepts all strings with any number of occurrences of 'a'. Answer: b

Explanation: The given alphabet contains only one symbol $\{a\}$ and the given NFA accepts all strings with any number of occurrences of 'a'. Answer: a

Explanation: Some are invalid because they both accept 'b' as a string which is not accepted by given DFA. D is invalid because it accepts bb^+a which are not accepted by given DFA. Answer: c

Explanation: We need minimum $n+1$ state to build NFA that accepts all substrings of a binary string. Following NFA accepts all strings containing substring of "010" and it has 4 states.

- a) String with substring 00**
- b) String with at most two 0's**
- c) String containing at least two 0's**
- d) None of the mentioned**

Answer: c
Explanation: The regular expression has two 0's surrounded by $(0+1)^$ which means accepted strings must have at least 2 0's.*

- 8. Which one of the following is FALSE?**
- a) Every NFA can be converted to DFA**
 - b) Every subset of a recursively enumerable set is recursive**
 - c) All of the mentioned**
 - d) None of the mentioned**

Answer: b
Explanation: Every subset of a recursively enumerable set is recursive.

- a) P-2, Q-1, R-3, S-4**
- b) P-1, Q-3, R-2, S-4**
- c) P-1, Q-2, R-3, S-4**
- d) P-3, Q-2, R-1, S-4**

Answer: c
Explanation: The above figures correctly represent the figures
*Eg: p gives $\epsilon + 0(01^*1 + 00)^*01^*$ and the q gives the expression $\epsilon + 0(10^*1 + 00)^*0$ and so on.*

- a) I and IV**
- b) I and III**
- c) I and only**
- d) IV**

Answer: a
Explanation: We can write DFA for both I and IV.
Answer: c
Explanation: Initially circle s around b so b^ then a then followed by $(a+b)^*$.*
Answer: b
Explanation: State q_0 can be omitted because it takes the same input as state q_1 hence by removing q_0 nothing changes.

Following is equivalent FSA with 2 states.

- 3. Which one of the following is TRUE?**

- a) The language $L = \{a^n b^n \mid n > 0\}$ is regular
- b) The language $L = \{a^n \mid n \text{ is prime}\}$ is regular
- c) The language $L = \{w \mid w \text{ has } 3k+1 \text{ b's for some } k\}$ is regular
- d) None of the mentioned

Answer: c

Explanation: Only for this option we can build a FA.

4.

- a) q_0, q_1, q_2
- b) q_0, q_1
- c) q_0, q_1, q_2, q_3
- d) q_3

Answer: a

Explanation: From q_0 state $\rightarrow 0$ then again remain on the same state 0, then to next state 1 and to q_2 we get 1. Answer: b

Explanation: $0^*1(1+00^*1)^*$

II) $0^*1^*1+11^*0^*1$

III) $(0+1)^*1$

(I) and (III) represent DFA.

(II) doesn't represent the DFA since it accepts strings like 11011, but the regular expression doesn't accept.

Answer: A

Explanation: L1 is regular let us considering the string 011011011011. Number of times 011 has occurred is 4 but also its occurrence is 3. Also if the string is ending with 011 we can make a 110. Now the next string: 110110110110 in this 110 has occurred 4 times and 011 3 times which already satisfy the . Answer: B

Explanation: baa is not regular so 3. Answer: b

Explanation: We can try some sample strings like aba, abbbabbb. Answer: c

Explanation: Language that can be accepted by DFA is 1001001 1001011, 1001101, 1001111, 1101001, 1111001, 1011001. Answer: b

Explanation: Either it takes 0 or 1 or iterations of it or none. [compilers-questions-answers-Obtaining the regular Expression from the Finite automata](#)

2. Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA is at least?

- a) N^2
- b) 2^N
- c) $2N$
- d) $N!$

Answer: c

Explanation: If the NFA has n states, the resulting DFA may have up to 2^n states, an exponentially larger number, which sometimes makes the construction impractical for large NFAs.

3. What can be said about a regular language L over {a} whose minimal finite state automaton has two states?

- a) L must be $\{a^n \mid n \text{ is odd}\}$
- b) L must be $\{a^n \mid n \text{ is even}\}$
- c) L must be $\{a^n \mid n \text{ is even}\}$
- d) Either L must be $\{a^n \mid n \text{ is odd}\}$, or L must be $\{a^n \mid n \text{ is even}\}$

Answer: d

Explanation: There are two states. When first state is final, it accepts even no. of a's. When second state is final, it accepts odd no. of a's.

4. How many minimum states are required to find whether a string has odd number of 0's or not?

- a) 1
- b) 2
- c) 3
- d) 4

Answer: b

Explanation: 2 minimum states are required to find whether a string has odd number of 0's or not.

Answer: b

Explanation: In DFA M: all strings must end with 'a'. In DFA N: all strings must end with 'b'. So the intersection is

empty. For an empty language, only one state is needed.

Which one of the following represents the strings in X^0 ?

- a) $10(0^* + (10)^*)1$
- b) $10(0^* + (10)^*)^*1$
- c) $10(0^* + (10)^*)^*1$
- d) $10(0 + 10)^*1 + 110(0 + 10)^*1$

Answer: c

Explanation: The smallest possible string by given grammar is "11".

$X^0 = 1X1$

$= 11X2$ [Replacing $X1$ with $1X2$]

$= 11$ [Replacing $X2$ with λ]

The string "11" is only possible with option $10(0^ + (10)^*)^*1$.*

- a) L1 and L3 only
- b) L2
- c) L2 and L3 only
- d) L3 only

Answer: a

Explanation: L3 is simple to guess, it is regular.

Below is DFA for L1.

L1 is interesting. The important thing to note about L1 is length of x is greater than 0.

8. The reorganizing capability of NFA and DFA is?

- a) May be different
- b) Must be different
- c) Must be same
- d) None of the mentioned

Answer: c

Explanation: Given any NFA one can construct an equivalent DFA.

9. Which of the following is not regular?

- a) String whose length is perfect square and consists of 0s
- b) Palindromes consisting of 0's and 1's
- c) String whose length is perfect square and consists of 0s & Palindromes consisting of 0's and 1's
- d) None of the mentioned

Answer: c

*Explanation: Strings of odd numbers of zeros can be generated by regular expression $(00)^*0$.*

10. Which of the following pairs of regular expression are equivalent?

- a) $1(01)^*$ and $(10)^*1$
- b) $X(xx)^*$ and $(xx)^*x$
- c) $1(01)^*$ and $(10)^*1$ & $X(xx)^*$ and $(xx)^*x$
- d) None of the mentioned

Answer: c

Explanation: $R1$ and $R2$ are reverse of each other. If any one of them can be generated then the other can be generated as well.

1. Which of the following identity is true?

- a) $\epsilon + RR^* = R^* = \epsilon + R^*R$
- b) $(R1R2)^*R1 = R1(R2R1)^*$
- c) $R^*R^* = R^*$
- d) All of the mentioned

Answer: d

Explanation: The former RE can be produced from the latter one.

2. The set of all strings over $\Sigma = \{a,b\}$ in which all strings having bbbb as substring is?

- a) $(a+b)^* bbbb (a+b)^*$
- b) $(a+b)^* bb (a+b)^* bb$
- c) $bbb(a+b)^*$
- d) $bb (a+b)^*$

Answer: a

Explanation: Out of all RE mentioned only the first string certainly has bbbb as substring. Rest all just have a possibility of having it.

3. The set of all strings over $\Sigma = \{a,b\}$ in which a single a is followed by any number of b's a single b followed by any number of a's is?

- a) $ab^* + ba^*$
- b) ab^*ba^*
- c) $a^*b + b^*a$
- d) None of the mentioned

Answer: a

Explanation: $ab^ + ba^*$ is the expression in which a single a is followed by any number of b's a single b followed by any number of a's.*

4. Regular expressions are used to represent which language?

- a) Recursive language
- b) Context free language
- c) Regular language
- d) All of the mentioned

Answer: c

Explanation: Regular expression is represented by regular language.

5. The set of all strings over $\Sigma = \{a,b\}$ in which strings consisting a's and b's and ending with in bb is?

- a) ab
- b) a^*bbb
- c) $(a+b)^*bb$
- d) All of the mentioned

Answer: c

Explanation: Only this expression ends with bb only.

6. If P, Q, R are three regular expressions and if P does not contain a then the equation $R = R + RP$ has a unique solution given by?

- a) $R = QP^*$
- b) $R = P^*Q$
- c) $R = RP$
- d) None of the mentioned

Answer: a

Explanation: It is an important law primarily used in conversion.

7. If L1 and L2 are regular languages is/are also regular language(s).

- a) $L1 + L2$
- b) $L1L2$
- c) $L1$
- d) All of the mentioned

Answer: d

Explanation: All these expression give us a regular grammar when L1 and L2 are regular.

8. Which of the following regular expression denotes zero or more instances of an a or b?

- a) a/b
- b) $(a/b)^*$
- c) $(ab)^*$
- d) a^*Ib

Answer: b

Explanation: This expression gives 0 or more instances of a or b.

9. The string $(a)((b)^*(c))$ is equivalent to _____

- a) Empty
- b) abcabc
- c) $b^*c|a$
- d) None of the mentioned

Answer: c

Explanation: Either b or a can lead followed by c this expression can be achieved by C as well.

1. In a compiler the module that checks every character of the source text is called _____

- a) The code generator
- b) The code optimizer

- c) The lexical analyzer
- d) The syntax analyzer

Answer: a

Explanation: Lexical analysis is the process of converting a sequence of characters into a sequence of tokens.

2. The context free grammar is ambiguous if _____

- a) The grammar contains non-terminals
- b) Produces more than one parse tree
- c) Production has two non-terminals side by side
- d) None of the mentioned

Answer: b

Explanation: Since more than one parse tree is generated hence one than option is available .Therefore it's ambiguous.

3. What is another name for Lexical Analyser?

- a) Linear Phase
- b) Linear Analysis
- c) Scanning
- d) All of the mentioned

Answer: d

Explanation: Lexical Analyzer is also called “Linear Phase” or “Linear Analysis” or “Scanning”.

4. An individual token is called _____

- a) Lexeme
- b) Lex
- c) Lexeme & Lex
- d) None of the mentioned

Answer: a

Explanation: Individual Token is also Called Lexeme.

Answer: a
Explanation: Lexical Analyzer's Output is given to Syntax Analysis.

6. Which phase of the compiler is Lexical Analyser?

- a) First
- b) Second
- c) Third
- d) None of the mentioned

Answer: a

Explanation: Lexical Analyzer is First Phase of Compiler.

7. Input to Lexical Analyser is _____

- a) Source Code
- b) Object Code
- c) Lexeme
- d) None of the mentioned

Answer: a

Explanation: Lexical analyser's Input is Source Code.

8. Lexical Analysis Identifies Different Lexical Units in a _____

- a) Source Code
- b) Object Code
- c) Lexeme
- d) None of the mentioned

Answer: a

Explanation: Lexical Analysis Identifies Different Lexical Units in a source Code.

9. Which one is a type of Lexeme?

- a) Identifiers
- b) Constants
- c) Keywords
- d) All of the mentioned

Answer: d

Explanation: All of them along with Operators are different types of lexemes.

10. A _____ is a string of characters which form a syntactic unit.

- a) Lexeme
- b) Lex
- c) Lexeme & Lex
- d) None of the mentioned

Answer: a

Explanation: A lexeme is a string of characters that form a syntactic unit.

1. The process of forming tokens from an input stream of characters is called _____

- a) Liberalisation
- b) Characterisation
- c) Tokenization
- d) None of the mentioned

Answer: c

Explanation: The process of forming tokens from an input stream of characters is called tokenization.

2. When expression sum=3+2 is tokenized then what is the token category of 3?

- a) Identifier
- b) Assignment operator
- c) Integer Literal
- d) Addition Operator

Answer: c

Explanation:

Lexeme

Token category

Sum	"Identifier"
=	"Assignment operator"
3	"Integer literal"
+	"Addition operator"
2	"Integer literal"
;	"End of statement".

3. Which grammar defines Lexical Syntax?

- a) Regular Grammar
- b) Syntactic Grammar
- c) Context free Grammar
- d) Lexical Grammar

Answer: d

Explanation: The specification of a programming language often includes a set of rules, the lexical grammar, which defines the lexical syntax.

4. Two Important lexical categories are _____

- a) White Space
- b) Comments
- c) None of the mentioned
- d) White Space & Comments

Answer: d

Explanation: Two important common lexical categories are white space and comments.

5. It has encoded within it information on the possible sequences of characters that can be contained within any of the tokens it handles. The mentioned function is performed by?

- a) Scanner
- b) Parser
- c) Syntactic Analyser
- d) All of the mentioned

Answer: a

Explanation: The first stage, the scanner, is FSM. It has encoded information on the possible sequences of characters that can be contained within any of the tokens it handles.

6. What goes over the characters of the lexeme to produce value?

- a) Scanner
- b) Parser
- c) Evaluator
- d) Lexical generator

Answer: a

Explanation: In order to construct a token, the lexical analyzer needs a second stage, the evaluator, which goes over the characters of the lexeme to produce a value.

Answer: a
Explanation: Lexers are often generated by a lexer generator same as parser.

8. Which one is a lexer Generator?

- a) ANTLR
- b) DRASTAR
- c) FLEX
- d) All of the mentioned

Answer: d

Explanation: ANTLR – Can generate lexical analyzers and parsers.

DFASTAR – Generates DFA matrix table-driven lexers in C++.

Flex – variant of the “lex” (C/C++).

Ragel – A state machine and lexer generator with output in C, C++, C#, Objective-C, D, Java, Go and Ruby.

9. Which of the lexical analyser can handle Unicode?

- a) Java CC
- b) JFLex
- c) Quex
- d) All of the mentioned

Answer: d

Explanation: JavaCC – JavaCC generates lexical analyzers written in Java.

JFLex – A lexical analyzer generator for Java.

Quex – A fast universal lexical analyzer generator for C and C++.

FsLex – A lexer generator for byte and Unicode character input for F#.

10. What is the output of a lexical analyzer?

- a) Machine Code
- b) Intermediate Code
- c) Stream of Token

d) Parse Tree

Answer: c

Explanation: The output given is in the form of tokens.

1. Which phase of the compiler is Syntax Analysis?

- a) First**
- b) Second**
- c) Third**
- d) None of the mentioned**

Answer: b

Explanation: It is Second Phase Of Compiler after Lexical Analyzer.

2. Syntax Analyser is also known as _____

- a) Hierarchical Analysis**
- b) Hierarchical Parsing**
- c) None of the mentioned**
- d) Hierarchical Analysis & Parsing**

Answer: d

Explanation: It is also called as Hierarchical Analysis or Parsing.

Answer: a
Explanation: It Groups Tokens of source Program into Grammatical Production.

4. From where it takes its input from?

- a) Lexical analyser**
- b) Syntactic Analyser**
- c) Semantic Analyser**
- d) None of the mentioned**

Answer: a

Explanation: A syntax analyzer or parser takes the input from a lexical analyzer in the form of token streams.

Answer: a
Explanation: Parsers are expected to parse the whole code even if some errors exist in the program.

6. A grammar for a programming language is a formal description of _____

- a) Syntax**
- b) Semantics**
- c) Structure**
- d) Library**

Answer: c

Explanation: The grammar clearly indicates which type of structure does a program has.

7. Which of these is not true about the Symbol Table?

- a) All the labels of the instructions are symbols**
- b) Table has entry for symbol name address value**
- c) Perform the processing of the assembler directives**
- d) Created during pass 1**

Answer: c

Explanation: The Symbol table does not ever perform the processing of the assembler derivative.

8. Which of these features of assembler are Machine-Dependent?

- a) Instruction formats**
- b) Addressing modes**
- c) Program relocation**
- d) All of the mentioned**

Answer: d

Explanation: All of these options are features of assembler which are machine dependent.

9. A compiler can check?

- a) Logical Error**
- b) Syntax Error**
- c) Both Logical and Syntax Error**
- d) Not Logical and Syntax Error**

Answer: b

Explanation: No compiler can ever check logical errors.

10. The fourth Generation computer was made up of _____

- a) Transistor**
- b) Vacuum tubes**
- c) Chips**
- d) Microprocessor chips**

Answer: d

Explanation: It is the only way to increase its throughput.

1. Select a Machine Independent phase of the compiler:

- a) Syntax Analysis**
- b) Intermediate Code generation**
- c) Lexical Analysis**
- d) All of the mentioned**

Answer: d

Explanation: All of them work independent of a machine.

2. A system program that combines the separately compiled modules of a program into a form suitable for execution?

- a) Assembler**
- b) Compiler**
- c) Linking Loader**
- d) Interpreter**

Answer: c

Explanation: A loader which combines the functions of a relocating loader with the ability to combine a number of program segments that have been independently compiled.

3. Which of the following system software resides in the main memory?

- a) Text Editor**
- b) Assembler**
- c) Linker**
- d) Loader**

Answer: d

Explanation: Loader is used to loading programs.

4. Output file of Lex is _____ the input file is Myfile.

- a) Myfile.e**
- b) Myfile.yy.c**
- c) Myfile.lex**
- d) Myfile.obj**

Answer: b

Explanation: This Produce the filr “myfile.yy.c” which we can then compile with g++.

5. Type checking is normally done during _____

- a) Lexical Analysis**
- b) Syntax Analysis**
- c) Syntax Directed Translation**
- d) Code generation**

Answer: c

Explanation: It is the function of Syntax directed translation.

Answer: a

Explanation: Syntax analyser does not check the type of the operand.

8. By whom is the symbol table created?

- a) Compiler**
- b) Interpreter**
- c) Assembler**
- d) None of the mentioned**

Answer: a

Explanation: Symbol table is created by the compiler which contains the list of lexemes or tokens.

9. What does a Syntactic Analyser do?

- a) Maintain Symbol Table**
- b) Collect type of information**
- c) Create parse tree**
- d) None of the mentioned**

Answer: c

Explanation: Syntax analyzer will just create a parse tree. Semantic Analyzer checks the meaning of the string parsed.

10. Semantic Analyser is used for?

- a) Generating Object code**
- b) Maintaining symbol table**
- c) Generating Object code & Maintaining symbol table**
- d) None of the mentioned**

Answer: c

Explanation: Maintaining the Symbol Table for each block.

Source Program for Semantic Errors.

Collects Type Information for Code Generation.

Reporting compile-time errors in the code Generating the object code (e.g., assembler or intermediate code).

Answer: a

Explanation: The proof of S1 can be seen in various book of theory of computation but s2 is a problem of category undecidable so a contradiction to this assumption can be easily obtained.

2. If P & R are regular and also given that if $PQ=R$, then?

- a) Q has to be regular**
- b) Q cannot be regular**
- c) Q need not be regular**
- d) Q has to be a CFL**

Answer: c

Explanation: If two regular languages when combined do not always produce a regular language.

3. Which of the following conversion is not possible (algorithmically)?

- a) Regular grammar to CFG**
- b) NDFA to DFA**
- c) NDPDA to DPDA**
- d) NDTM to DTM**

Answer: c

Explanation: Not every NDPDA has an equivalent deterministic PDA.

4. Consider the grammar given below $E \rightarrow E+E \mid E * E \mid E-E \mid E/E \mid E^E \mid (E) \mid id$. Assume that + and ^ have the same but least precedence, * and / have the next higher precedence but the same precedence and finally ^ has the highest precedence. Assume + and ^ associate to the left like * and / and that ^ associates to the right. Choose the correct for the ordered pairs (^, ^), (-, -), (+, +), (*, *) in the operator precedence table constructed for the grammar:

- a) All <
- b) All >
- c) < >, =
- d) < > > >

Answer: d

Explanation: This relation is established on the basis of the precedence of operators.

5. Recursively enumerable languages are not closed under _____

- a) Union
- b) Intersection
- c) Complementation
- d) Concatenation

Answer: c

Explanation: Recursive languages are closed under the following operations.

The Kleene star L^* of L

the concatenation $L \circ P$ of L and P

the union $L \cup P$

the intersection $L \cap P$.

6. Grammar that produce more than one Parse tree for same sentence is _____

- a) Ambiguous
- b) Unambiguous
- c) Complementation
- d) Concatenation Intersection

Answer: a

Explanation: An ambiguous grammar is one for which there is more than one parse tree for a single sentence.

7. Automaton accepting the regular expression of any number of a's is _____

- a) a^*
- b) ab^*
- c) $(a/b)^*$
- d) a^*b^*c

Answer: a

Explanation: It gives any number of a's.

8. Grammars that can be translated to DFAs is _____

- a) Left linear grammar
- b) Right linear grammar
- c) Generic grammar
- d) All of the mentioned

Answer: b

Explanation: Right linear grammar can be translated to the DFAs.

9. Which of the following language accepted by a Push down Automata?

- a) Type 0
- b) Type 1

- c) Type2
- d) Type3

Answer: c

Explanation: A known fact that type 2 grammar is accepted by PDA.

10. Given the following statements: (i) Recursive enumerable sets are closed under complementation. (ii) Recursive sets are closed under complements. Which is/are the correct statements?

- a) I only
- b) II only
- c) Both I and II
- d) Neither I nor II

Answer: b

Explanation: Recursive languages are closed under the following operations.

The Kleene star L^* of L

The concatenation $L \circ P$ of L and P

The union $L \cup P$

The intersection $L \cap P$.

1. Assume statements S1 and S2 defined as: S1: $L_2 - L_1$ is recursive enumerable where L_1 and L_2 are recursive and recursive enumerable respectively. S2: The set of all Turing machines is countable. Which of the following is true?

- a) S1 is correct and S2 is not correct
- b) Both S1 and S2 are correct
- c) Both S1 and S2 are not correct
- d) S1 is not correct and S2 is correct

Answer: b

Explanation: The assumptions of statement S1 and S2 are correct.

2. A context free language is called ambiguous if _____

- a) It has 2 or more left derivations for some terminal string $w \in L(G)$
- b) It has 2 or more right derivations for some terminal string $w \in L(G)$
- c) It has 2 or more left & right derivations for some terminal string $w \in L(G)$
- d) None of the mentioned

Answer: b

Explanation: A context-free grammar (CFG) is a set of recursive rewriting rules (or productions) used to generate patterns of strings.

3. Which of the following statement is false?

- a) The CFG can be converted to Chomsky normal form
- b) The CFG can be converted to Greibach normal form
- c) CFG is accepted by pushdown automata
- d) None of the mentioned

Answer: d

Explanation: All the statements follow the rules.

4. The context free grammar $S \rightarrow A111|S1$, $A \rightarrow A0 | 00$ is equivalent to _____

- a) $\{0^n 1^m \mid n=2, m=3\}$
- b) $\{0^n 1^m \mid n=1, m=5\}$
- c) $\{0^n 1^m \mid n \text{ should be greater than two and } m \text{ should be greater than four}\}$
- d) None of the mentioned

Answer: a

Explanation: $S \rightarrow A111$

$S \rightarrow 00111$ ($A \rightarrow 00$).

5. The context free grammar $S \rightarrow SS \mid 0S1 \mid 1S0 \mid \epsilon$ generates _____

- a) Equal number of 0's and 1's
- b) Unequal number of 0's and 1's
- c) Number of 0's followed by any number of 1's
- d) None of the mentioned

Answer: a

Explanation: $S \rightarrow SS$

$S \rightarrow 0S1S$

$S \rightarrow 0S11S0$

$S \rightarrow 0110$.

6. Which of the following statement is false?

- a) In derivation tree, the label of each leaf node is terminal
- b) In derivation tree, the label of all nodes except leaf nodes is a variable
- c) In derivation tree, if the root of a sub tree is X then it is called –tree
- d) None of the mentioned

Answer: d

Explanation: All of them are true regarding a derivation tree.

7. Push down automata accepts which language?

- a) Context sensitive language
- b) Context free language
- c) Recursive language
- d) None of the mentioned

Answer: b

Explanation: PDA accepts CFG.

8. A regular Grammar is a _____

- a) CFG
- b) Non CFG
- c) English Grammar
- d) None of the mentioned

Answer: a

Explanation: Regular grammar is CFG. It restricts its rules to a single non terminal on left hand side.

9. A CFG is closed under _____

- a) Union
- b) Kleene star
- c) Concatenation
- d) None of the mentioned

Answer: d

Explanation: CFG is closed under the above mentioned 3 operations.

10. Which of these does not belong to CFG?

- a) Terminal Symbol
- b) Non terminal Symbol
- c) Start symbol
- d) End Symbol

Answer: d

Explanation: CFG consists of terminal non terminal start symbol set of production rules but does not have an end

symbol.

- 1. How many strings of length less than 4 contains the language described by the regular expression $(x+y)^*y(a+ab)^*$?**
- a) 7**
 - b) 10**
 - c) 12**
 - d) 11**

Answer: d

Explanation: String of length 0 = 1

string of length 1 = 4

string of length 2 = 3

string of length 3 = 3.

- 2. Which of the following is true?**

- a) $(01)^*0 = 0(10)^*$**
- b) $(0+1)^*0(0+1)^*1(0+1) = (0+1)^*01(0+1)^*$**
- c) $(0+1)^*01(0+1)^*+1^*0^* = (0+1)^*$**
- d) All of the mentioned**

Answer: d

Explanation: None.

- 3. A language is regular if and only if?**

- a) Accepted by DFA**
- b) Accepted by PDA**
- c) Accepted by LBA**
- d) Accepted by Turing machine**

Answer: a

Explanation: All of above machine can accept regular language but all string accepted by machine is regular only for DFA.

- 4. What is Regular grammar?**

- a) Context free grammar**
- b) Non context free grammar**
- c) English grammar**
- d) None of the mentioned**

Answer: a

Explanation: Regular grammar is subset of context free grammar.

- 5. Let the class of language accepted by finite state machine be L_1 and the class of languages represented by regular expressions be L_2 then?**

- a) $L_1 < L_2$**
- b) $L_1 \geq L_2$**
- c) $L_1 \cup L_2 = .^*$**
- d) $L_1 = L_2$**

Answer: d

Explanation: Finite state machine and regular expression have same power to express a language.

- 6. Which of the following is not a regular expression?**

- a) $[(a+b)^*-(aa+bb)]^*$**
- b) $[(0+1)-(0b+a1)^*(a+b)]^*$**
- c) $(01+11+10)^*$**
- d) $(1+2+0)^*(1+2)^*$**

Answer: b

Explanation: Except $[(0+1)-(0b+a1)^*(a+b)]^*$ all are regular expression.

7. Regular expression is _____

- a) Type 0 language
- b) Type 1 language
- c) Type 2 language
- d) Type 3 language

Answer: a

Explanation: According to the Chomsky hierarchy.

8. Which of the following is true?

- a) All subsets of a regular set are always regular
- b) All finite subsets of non-regular set are always regular
- c) Union of two non regular set of language is not regular
- d) Infinite times union of finite set is always regular

Answer: b

Explanation: None.

9. L and $\sim L$ are recursive enumerable then L is?

- a) Regular
- b) Context free
- c) Context sensitive
- d) Recursive

Answer: d

Explanation: If L is recursive enumerable and its complement too if and only if L is recursive.

10. Regular expressions are closed under _____

- a) Union
- b) Intersection
- c) Kleene star
- d) All of the mentioned

Answer: d

Explanation: According to the definition of regular expression.

1. Give a production grammar that specified language $L = \{a^i b^{2i} \mid i \geq 1\}$.

- a) $\{S \rightarrow aSbb, S \rightarrow abb\}$
- b) $\{S \rightarrow aSb, S \rightarrow b\}$
- c) $\{S \rightarrow aA, S \rightarrow b, A \rightarrow b\}$
- d) None of the mentioned

Answer: a

Explanation: $S \rightarrow aSbb, S \rightarrow abb$ give $a^i b^{2i}$ a's and b^{2i} b's.

2. Let R1 and R2 be regular sets defined over alphabet Σ then?

- a) $R1 \cup R2$ is regular
- b) $R1 \cap R2$ is regular
- c) $\Sigma \cap R2$ IS NOT REGULAR
- d) $R2^*$ IS NOT REGULAR

Answer: a

Explanation: Union of 2 regular languages is regular.

3. Which of the following String can be obtained by the language $L = \{a^i b^{2i} \mid i \geq 1\}$?

- a) aaabbbbbbb
- b) aabbbb
- c) abbabbba
- d) aaaabbbabb

Answer: a

Explanation: Above production rule gives suppose if 3 a’s the corresponding b’s are 6 b’s.

4. Give a production grammar for the language $L = \{x/x \in (a,b)^*, \text{ the number of a’s in } x \text{ is multiple of } 3\}$.
- a) $\{S \rightarrow bS, S \rightarrow b, S \rightarrow aA, S \rightarrow bA, A \rightarrow aB, B \rightarrow bB, B \rightarrow aS, S \rightarrow a\}$
 - b) $\{S \rightarrow aS, S \rightarrow bA, A \rightarrow bB, B \rightarrow bBa, B \rightarrow bB\}$
 - c) $\{S \rightarrow aaS, S \rightarrow bbA, A \rightarrow bB, B \rightarrow ba\}$
 - d) None of the mentioned

Answer: a

Explanation: The above given condition is satisfied by

$S \rightarrow bS \ S \rightarrow B$
 $S \rightarrow aA \ s \rightarrow bA \ A \rightarrow aB \ B \rightarrow bB$
 $B \rightarrow aS \ S \rightarrow a.$

5. The production Grammar is $\{S \rightarrow aSbb, S \rightarrow abb\}$ is?
- a) type-3 grammar
 - b) type-2 grammar
 - c) type-1 grammar
 - d) type-0 grammar

Answer: b

Explanation: Type 2 grammar satisfies this production grammar.

6. The regular expression denote a language comprising all possible strings of even length over the alphabet (0,1) is?
- a) $1 + 0(1+0)^*$
 - b) $(0+1)(1+0)^*$
 - c) $(1+0)$
 - d) $(00+0111+10)^*$

Answer: d

Explanation: The condition is satisfied by 00 or 0111 or 10 or iterations of these. Answer: a

Explanation: The above stated grammar is non-linear because it has two non-terminals on the right-hand side.

- a) Linear
- b) Nonlinear
- c) Linear & Nonlinear
- d) None of the mentioned

Answer: a

Explanation: Grammar is non-linear because one of the rules (the first one) has two non-terminals on the right-hand side. Answer: a

Explanation: Grammar is linear because no rule has more than one non terminal on the right-hand side. Answer: a
Explanation: Right-Linear grammars, following are the form of productions: $A \rightarrow xB$ or $A \rightarrow x$ where x is some string of terminals.

- a) Right Linear Grammar
- b) Left Linear Grammar
- c) Right & Left Linear Grammar
- d) None of the mentioned

Answer: a

Explanation: grammars in which all of the rules contain only one non-terminal on the left-hand side, and where in every

case that non-terminal is the first symbol are called right Linear.

6. What are the two types of Linear Grammar?

- a) Right Linear**
- b) Left Linear**
- c) None of the mentioned**
- d) Right & Left Linear**

Answer: d

Explanation: Linear grammar is of 2 types Left and Right Linear Grammar

7. Which Grammar is it?

- a) Right Linear**
- b) Left Linear**
- c) None of the mentioned**
- d) Right & Left Linear**

Answer: b

Explanation: In Left-Linear grammars, all productions have the form: $A \rightarrow Bx$ or $A \rightarrow x$ where x is some string of terminals.

- a) Right Linear**
- b) Left Linear**
- c) None of the mentioned**
- d) Right & Left Linear**

Answer: b

*Explanation: In this case they both correspond to the regular expression $(ab)^*a$.*

Answer: a
Explanation: As it turns out the languages that can be generated by Regular Grammars is equivalent to those that can be specified by Regular Expressions.

Answer: a
Explanation: That's why they are called regular languages.

Answer: a
Explanation: Since right-linear grammars are regular, it follows that left-linear grammars are also regular.

2. What is CFG?

- a) Compiler**
- b) A language expression**
- c) Regular Expression**
- d) None of the mentioned**

Answer: b

Explanation: They are defined by rule $A \rightarrow b$ where A is non terminal and b is terminal.

3. What is the idea of automation with a stack as auxiliary storage?

- a) Finite automata**
- b) Push Down Automata**
- c) Deterministic Automata**
- d) None of the mentioned**

Answer: b

Explanation: Push Down Automata manipulate the Stack as a part of performing a transition.

4. Transition of finite automata is _____

- a) Finite Diagram**
- b) State Diagram**
- c) Node Diagram**
- d) E-R Diagram**

Answer: b

Explanation: Transition of finite automata is Finite Diagram.

5. A context free language is called ambiguous if?

- a) It has 2 or more than 2 left derivations for some terminal string $w \in L(G)$
- b) It has 2 or more than 2 right derivations for some terminal string $w \in L(G)$
- c) It has 2 or more than 2 left and right derivations for some terminal string $w \in L(G)$
- d) None of the mentioned

Answer: c

Explanation: When two or more Left and right most derivative occur the grammar turn ambiguous .

6. Which of the following statement is true?

- a) Every language that is defined by regular expression can also be defined by finite automata
- b) Every language defined by finite automata can also be defined by regular expression
- c) We can convert regular expressions into finite automata
- d) All of the mentioned

Answer: d

Explanation: All these statements are true w.r.t regular expression.

7. Which of the following identity is wrong?

- a) $R + R = R$
- b) $(R^*)^* = R^*$
- c) $\epsilon R = R\epsilon = R$
- d) $\emptyset R = R\emptyset = RR^*$

Answer: d

Explanation: Regular grammar combined with empty does not give R^ instead gives empty.*

8. Grammars that can be translated to DFAs is _____

- a) Left linear grammar
- b) Right linear grammar
- c) Generic grammar
- d) All of the mentioned

Answer: b

Explanation: Right Linear grammar can be translate to DFA.

9. A language is regular if and only if it is accepted by finite automata.

- a) The given statement statement is true
- b) Given statement is false
- c) Statement is partially true
- d) None of the mentioned

Answer: a

Explanation: Regular Language is accepted by Finite Automata. Every regular language is Context free.

10. A Push Down Automata is if there is at most one transition applicable to each configuration?

- a) Deterministic
- b) Non deterministic
- c) Finite
- d) Non finite

Answer: a

Explanation: In every situation, only one transition is available as continuation then the result is deterministic push down automata.

1. Which of the following derivations does a top-down parser use while parsing an input string?

- a) Leftmost derivation
- b) Leftmost derivation in reverse
- c) Rightmost derivation
- d) Rightmost derivation in reverse

Answer: a

Explanation: In top down parser takes input from Left to right constructing leftmost derivation of the sentence.

2. The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called?

- a) Assembly
- b) Parsing
- c) Relocation
- d) Symbol resolute

Answer: c

Explanation: Relocation is the process of replacing symbolic references or names of libraries with actual usable addresses in memory before running a program. Linker performs it during compilation.

3. Which of the following statements is false?

- a) Left as well as right most derivations can be in Unambiguous grammar
- b) An LL (1) parser is a top-down parser
- c) LALR is more powerful than SLR
- d) Ambiguous grammar can't be LR (k)

Answer: a

Explanation: If a grammar has more than one leftmost (or rightmost) derivation the grammar is ambiguous. Sometimes in unambiguous grammar the rightmost derivation and leftmost derivations may differ.

- a) (i) only
- b) (i) and (iii) only
- c) (ii) and (iii) only
- d) (iii) and (iv) only

Answer: b

Explanation: An operator precedence parser is a bottom-up parser that interprets an operator-precedence grammar. Consider the grammar with the following translation rules and E as the start symbol.

$A \rightarrow A1 \# B \{A.value = A1.value * B.value\}$

$| B \{A.value = B.value\}$

$B \rightarrow B1 \& F \{B.value = B1.value + C.value\}$

$| C \{B.value = C.value\}$

$C \rightarrow num \{C.value = num.value\}.$

5. Compute E.value for the root of the parse tree for the expression: $2 \# 3 \& 5 \# 6 \& 4.$

- a) 200
- b) 180
- c) 160
- d) 40

Answer: c

Explanation: Higher precedence operator will never produce an expression with operator with lower precedence. $\& \> \#$ in terms of precedence order.

which of the following is true?

- a) * has higher precedence than +
- b) – has higher precedence than *
- c) + and — have same precedence
- d) + has higher precedence than *

Answer: b

Explanation: Precedence is that a higher precedence operator will never produce an expression with operator with lower precedence.

In the given grammar MINUS has higher precedence than ASTERIX.

7. Consider a program P that consists of two source modules M1(contains reference to a function defined in M2) and M2 contained in two different files.

- a) Edit time**
- b) Compile time**
- c) Link time**
- d) Load time**

Answer: c

Explanation: Compiler transforms source code into the machine language which is in binary.

Kinds of object codes:

- i. Defined symbols, which allow it to be called by other modules,*
- ii. Undefined symbols, which call the other modules where these symbols are defined, and*
- iii. Symbols which are used internally within object file for relocation.*

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

- a) Removing left recursion only**
- b) Factoring the grammar alone**
- c) Factoring & left recursion removal**
- d) None of the mentioned**

Answer: d

Explanation: Factoring as well as left recursion removal do not suffice to convert an arbitrary CFG to LL(1) grammar.

9. Assume that the SLR parser for a grammar G has n1 states and the LALR parser for G has n2 states.

- a) n1 is necessarily less than n2**
- b) n1 is necessarily equal to n2**
- c) n1 is necessarily greater than n2**
- d) none of the mentioned**

Answer: b

Explanation: SLR parser has less range of context free languages than LALR but still both n1 & n2 are same for SLR & LALR respectively.

10. Match the following.

P. Regular expression	1. Syntax analysis
Q. Pushdown automata	2. Code generation
R. Dataflow analysis	3. Lexical analysis
S. Register allocation	4. Code optimization

- a) P-4, Q-1, R-2, S-3**
- b) P-3, Q-1, R-4, S-2**
- c) P-3, Q-4, R-1, S-2**
- d) P-2, Q-1, R-4, S-3**

Answer: b

Explanation: Syntax analysis has Regular expressions. The code optimization goes hand in hand with data flow analysis. Whereas CFG is related to PDA which is related to syntax analysis Register allocation is used in reference with code generation.

- a) I and II**
- b) I and IV**
- c) III and IV**
- d) I III and IV**

Answer: b

Explanation: In recursion, space used but recursive call can't be calculated by the compiler.

2. Which of the following describes a handle (as applicable to LR-parsing) appropriately?

- a) Position where next reduce or shift operation will occur
- b) The next step has use of Non-terminal for reduction
- c) Used for reduction in a coming-up step along with a position in the sentential form where the next shift or reduce operation will occur
- d) Used in the next step for reduction along with a position in the sentential form where the right hand side of the production may be found

Answer: d

Explanation: the next step in LR parsing shall have a Reduction.

3. Which one of the following is a top-down parser?

- a) Recursive descent parser
- b) Operator precedence parser
- c) An LR(k) parser
- d) An LALR(k) parser

Answer: a

Explanation: Recursive Descent also known as top down parsing also known to be LL(1).

Consider the following two statements:

P: Every regular grammar is LL(1)

Q: Regular is LR(1) grammar.

4. Which of the following is TRUE?

- a) Both P and Q are true
- b) P is true and Q is false
- c) P is false and Q is true
- d) Both P and Q are false

Answer: c

Explanation: Ambiguity can be seen in regular grammar

$S \rightarrow aA/a$

$A \rightarrow aA/\epsilon$

In above grammar, string 'a' has two leftmost derivations.

$S \rightarrow aA$

$S \rightarrow a$

$S \rightarrow a$ (using $A \rightarrow \epsilon$).

5. Consider the grammar defined by the following production rules:

```
S --> T * P
T --> U | T * U
P --> Q + P | Q
Q --> Id
U --> Id
```

Which one of the following is TRUE?

- a) + is left associative, while * is right associative
- b) + is right associative, while * is left associative
- c) Both + and * are right associative
- d) Both + and * are left associative

Answer: b

Explanation: It is associative we can see and tell.

Second productions latter part shows left recursion and is left associative.

6. The grammar $A \rightarrow AA \mid (A) \mid \epsilon$ is not suitable for predictive-parsing because the grammar is?

- a) Ambiguous
- b) Left recursive
- c) Right recursive
- d) An operator grammar

Answer: b

Explanation:

$$A ::= A a$$

| b is the left recursive language.

For a sentence $n + n \times n$, the handles in the right-sentential form of the reduction are _____

- a) n, E + n and E + n × n
- b) n, E + n and E + n × n
- c) n, n + n and n + n × n
- d) n, E + n and E × n

Answer: d

Explanation: $E \rightarrow E + n$ {Applying $E \rightarrow E + n$ }

$\rightarrow E + E * n$ {Applying $E \rightarrow E * n$ }

$\rightarrow E + n * n$ {Applying $E \rightarrow n$ }

$\rightarrow n + n * n$ {Applying $E \rightarrow n$ }.

8. Which grammar rules violate the requirements of an operator grammar?

- 1. $P \rightarrow Q R$
- 2. $P \rightarrow Q s R$
- 3. $P \rightarrow \epsilon$
- 4. $P \rightarrow Q t R r$

- a) 1 only
- b) 1 and 3 only
- c) 2 and 3 only
- d) 3 and 4 only

Answer: b

Explanation: Top down parsin: We begin with the start symbol and compare the right side of the different productions against the first piece of input to see which of the productions should be used.

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

- a) Removing left Recursive alone
- b) Factoring the grammar alone
- c) Along with removing left recursion we also perform the factoring of the grammar
- d) None of the mentioned

Answer: d

Explanation: Removing left recursion and factoring the grammar do not suffice to convert an arbitrary CFG to LL(1) grammar.

10. In a bottom-up evaluation of a syntax directed definition its inherited attributes can do which of the following?

- a) Always evaluated
- b) Can be evaluated if the definition is L attributed
- c) Can be evaluated if the definition has synthesized attributes
- d) Never be evaluated

Answer: b

Explanation: A Syntax Directed Definition (SDD) is called S Attributed if it has only synthesized attributes. Also the L-Attributed Definitions contain both synthesized and inherited attributes but do not need to build a dependency graph to evaluate them.

- a) LL(1)
- b) SLR(1) but not LL(1)
- c) LALR(1) but not SLR(1)
- d) LR(1) but not LALR(1)

Answer: a

Explanation: Since there is no conflict, the grammar is LL (1) hence a predictive parse table with no conflicts can be constructed.

2. Which of the following statements is false?
- a) Unambiguous grammar has both kind of derivations
 - b) An LL(1) parser is a top-down parser
 - c) LALR is more powerful than SLR
 - d) Ambiguous grammar can't be LR(k)

Answer: a

Explanation: If a grammar has more than one leftmost (or rightmost) derivation the grammar is ambiguous.

- Which of the following is true?
- a) * has higher precedence than +
 - b) – has higher precedence than *
 - c) + and — have same precedence
 - d) + has higher precedence than *

Answer: b

Explanation: e.g. input is 3*4-5 r



First ‘ - ‘ is be evaluated then ‘ * ’

4. Which one of the following is true at any valid state in shift-reduce parsing?
- a) At the bottom we find the prefixes
 - b) None of the mentioned
 - c) Stack contains only viable prefixes
 - d) Stack consists of viable prefixes

Answer: c

Explanation: The prefixes on the stack of a shift-reduce parser are called viable prefixes.

5. In the context of abstract-syntax-tree and control-flow-graph. Which one of the following is true?
- a) In both AST and CFG if node N2 be the successor of node N1
 - b) For any input program, neither AST nor CFG will contain a cycle
 - c) The max number of successors of a node in an AST and a CFG depends on the input program
 - d) None of the mentioned

Answer: c

Explanation: Successors depends on input .

6. Match the following.

List-I	List-II
A. Lexical analysis	1. Graph coloring
B. Parsing	2. DFA minimization

- a) A – 2, B – 3, C – 1, D – 4
- b) A – 2, B – 1, C – 4, D – 3
- c) A – 2, B – 4, C – 1, D – 3
- d) A – 2, B – 3, C – 4, D – 1

Answer: c

Explanation: The entire column an items matches the Column B items in a certain way.

7. Which of the following pairs is the most powerful?

- a) SLR, LALR
- b) Canonical LR ,LALR
- c) SLR canonical LR
- d) LALR canonical LR

Answer: c

Explanation parser algorithm is simple.

8. Consider the following grammar G.

S → F		H
F → p		c
H → d		c

Answer: d

Explanation: There is ambiguity as the string can be derived in 2 possible ways.

First Leftmost Derivation

$S \rightarrow F$

$F \rightarrow c$

Second Leftmost Derivation

$S \rightarrow H$

$H \rightarrow c.$

9. What is the maximum number of reduce moves that can be taken by a bottom-up parser for a grammar with no epsilon- and unit-production to parse a string with n tokens?

- a) $n/2$
- b) $n-1$
- c) $2n-1$
- d) 2^n

Answer: b

Explanation: The moves are $n-1$.

10. Consider the following two sets of LR (1) items of an LR (1) grammar:

X → c.X, c/d
X → .cX, c/d
X → .d, c/d
X → c.X, \$
X → .cX, \$
X → .d, \$

Which one is false?

- 1. Cannot be merged since look ahead's are different.
- 2. Can be merged but will result in S-R conflict.
- 3. Can be merged but will result in R-R conflict.
- 4. Cannot be merged since goto on c will lead to two different sets.

- a) 1 only

- b) 2 only
- c) 1 and 4 only
- d) 1, 2, 3 and 4 only

Answer: d

Explanation: All these are valid reasons.

1. Inherited attribute is a natural choice in _____

- a) Variable declarations record is maintained
- b) L values and R values
- c) All of the mentioned
- d) None of the mentioned

Answer: a

Explanation: It keeps track of variable.

2. YACC builds up _____

- a) SLR parsing table
- b) Canonical LR parsing table
- c) LALR parsing table
- d) None of the mentioned

Answer: c

Explanation: It is a parser generator.

3. In an absolute loading scheme which loader function is accomplished by assembler?

- a) Re-allocation
- b) Allocation
- c) Linking
- d) Loading

Answer: a

Explanation: Large number variables onto a small number of CPU register.

4. A parser with the valid prefix property is advantageous because it _____

- a) Detects errors
- b) None of the mentioned
- c) Errors are passed to the text phase
- d) All of the mentioned

Answer: c

Explanation: Advantage for a valid prefix property.

5. The action of parsing the source program into proper syntactic classes is called _____

- a) Syntax Analysis
- b) Lexical Analysis
- c) Interpretation analysis
- d) General Syntax Analysis

Answer: b

Explanation: Conversion of characters to tokens.

6. Relocating bits used by relocating loader are specified by _____

- a) Relocating loader itself
- b) Linker
- c) Assembler
- d) Macro Processor

Answer: b

Explanation: Takes an object files and combines them into a single executable file, library file, or another object file.

7. What is the binary equivalent of the decimal number 368?

- a) 10111000**
- b) 110110000**
- c) 111010000**
- d) 111100000**

Answer: b

Explanation: 368 binary equivalents is

8=1000

6=0110

3=0011

So 1101101000.

8. $AB+(A+B)'$ is equivalent to _____

- a) $A \cdot B$**
- b) $A+B$**
- c) $(A+B)A$**
- d) $(A+B)B$**

Answer: a

Explanation: It is equivalent to $A \cdot B$.

9. A top down parser generates _____

- a) Rightmost Derivation**
- b) Right most derivation in reverse**
- c) Left most derivation**
- d) Left most derivation in reverse**

Answer: c

Explanation: Top-down parsing is a parsing strategy where one first looks at the highest level of the parse tree and works down the parse tree by using the rewriting rules of a formal grammar.

10. Running time of a program depends on _____

- a) Addressing mode**
- b) Order of computations**
- c) The usage of machine idioms**
- d) All of the mentioned**

Answer: d

Explanation: Run time, runtime or execution time is the time during which a program is running (executing).

1. Which of the following is the fastest logic?

- a) TTL**
- b) ECL**
- c) CMOS**
- d) LSI**

Answer: b

Explanation: In electronics, emitter-coupled logic (ECL) is a high-speed integrated circuit.

2. A bottom up parser generates _____

- a) Right most derivation**
- b) Rightmost derivation in reverse**
- c) Leftmost derivation**
- d) Leftmost derivation in reverse**

Answer: b

Explanation: This corresponds to starting at the leaves of the parse tree also known as shift-reduce parsing.

3. A grammar that produces more than one parse tree for some sentence is called _____

- a) Ambiguous**
- b) Unambiguous**
- c) Regular**
- d) None of the mentioned**

Answer: a

Explanation: ambiguous grammar has more than one parse tree.

4. An optimizer Compiler _____

- a) Is optimized to occupy less space**
- b) Is optimized to occupy less space & Optimize the code**
- c) Optimize the code**
- d) None of the mentioned**

Answer: d

Explanation: In computing, an optimizing compiler is a compiler that tries to minimize or maximize some attributes of an executable computer program.

5. The linker _____

- a) Is similar to interpreter**
- b) Uses source code as its input**
- c) Is required to create a load module**
- d) None of the mentioned**

Answer: c

Explanation: It is a program that takes one or more object files generated by a compiler and combines them into a single executable file, library file, or another object file.

6. A latch is constructed using which two cross coupled?

- a) AND OR gates**
- b) AND gates**
- c) NAND and NOR gates**
- d) NAND gates**

Answer: d

Explanation: It has two inputs and one output.

7. Pee Hole optimization _____

- a) Loop Optimization**
- b) Local Optimization**
- c) Constant folding**
- d) Data Flow analysis**

Answer: c

Explanation: More loops are added.

8. The optimization which avoids test at every iteration is?

- a) Loop unrolling**
- b) Loop jamming**
- c) Constant folding**
- d) None of the mentioned**

Answer: a

Explanation: Execution speed is enhanced by sacrificing bits.

9. Scissoring enables _____

- a) A part of data to be displayed**
- b) Entire data to be displayed**
- c) None of the mentioned**
- d) No data to be displayed**

Answer: a

Explanation: Displays only some part of the data.

10. Shift reduce parsers are _____

- a) Top down Parser**
- b) Bottom Up parser**
- c) May be top down or bottom up**
- d) None of the mentioned**

Answer: b

Explanation: Also known as shift reduce parser.

1. DAG representation of a basic block allows _____

- a) Automatic detection of local common sub expressions**
- b) Detection of induction variables**
- c) Automatic detection of loop variant**
- d) None of the mentioned**

Answer: a

Explanation: It detects local sub expression.

2. Inherited attribute is a natural choice in _____

- a) Tracking declaration of a variable**
- b) Correct use of L and R values**
- c) All of the mentioned**
- d) None of the mentioned**

Answer: a

Explanation: These attribute keep a check on variable declaration.

3. An intermediate code form is _____

- a) Postfix notation**
- b) Syntax Trees**
- c) Three Address code**
- d) All of the mentioned**

Answer: d

Explanation: Intermediate code generator receives input from its predecessor phase, semantic analyzer, in the form of an annotated syntax tree.

4. Which of the following actions an operator precedence parser may take to recover from an error?

- a) Insert symbols onto the stack**
- b) Delete symbols from the stack**
- c) Inserting or deleting symbols from the input**
- d) All of the mentioned**

Answer: d

Explanation: All these symbols are used to recover operator precedence parser from an error.

5. What is the output of lexical analyzer?

- a) A set of regular expression**
- b) Syntax tress**

- c) Set of Token
- d) String of Characters

Answer: c

Explanation: Lexical analysis is the process of converting a sequence of characters into a sequence of tokens.

6. Which of the following is used for grouping of characters into tokens?

- a) Parser
- b) Code optimization
- c) Code generator
- d) Lexical analyser

Answer: d

Explanation: Lexical analysis is the process of converting a sequence of characters into a sequence of tokens.

7. Shift reduce parsers are _____

- a) Top down parser
- b) Bottom up parser
- c) Maybe both
- d) None of the mentioned

Answer: b

Explanation: This corresponds to starting at the leaves of the parse tree. It can be thought of a process of reducing the string in question to the start symbol of the grammar. Bottom-up parsing is also known as shift-reduce parsing.

8. A bottom up parser generates _____

- a) Right most derivation
- b) Right most derivation in reverse
- c) Left most derivation
- d) Left most derivation in reverse

Answer: b

Explanation: This corresponds to starting at the leaves of the parse tree. It can be thought of. A process of reducing the string in question to the start symbol of the grammar. Bottom-up parsing is also known as shift-reduce parsing.

9. What is garbage?

- a) Unallocated storage
- b) Allocated storage whose access paths are destroyed?
- c) Allocated storage
- d) Uninitialized storage

Answer: b

Explanation: These are more like memory locations with values whose pointers have been revoked.

10. An optimizing compiler _____

- a) Is optimized to occupy less space
- b) Is optimized to take less time for execution
- c) Optimized the code
- d) None of the mentioned

Answer: c

Explanation: As the name suggests that it optimizes the code.

1. Input to code generator is _____

- a) Source code
- b) Intermediate code
- c) Target code
- d) All of the mentioned

Answer: b

Explanation: Intermediate code is the input to the code generator.

2. A synthesized attribute is an attribute whose value at a parse tree node depends on _____

- a) Attributes at the siblings only**
- b) Attributes at parent node only**
- c) Attributes at children nodes only**
- d) None of the mentioned**

Answer: c

Explanation: Synthesized attribute's value depend on children node only.

3. In a bottom up evaluation of a syntax direction definition, inherited attributes can _____

- a) Always be evaluated**
- b) Be evaluated only if the definition is L-attributed**
- c) Evaluation only done if the definition has synthesized attributes**
- d) None of the mentioned**

Answer: c

Explanation: Bottom-up parsing identifies and processes the text's lowest-level, before its mid-level structures, and the highest-level overall structure to last are left.

4. The graph that shows basic blocks and their successor relationship is called _____

- a) DAG**
- b) Flow Chart**
- c) Control Graph**
- d) Hamilton graph**

Answer: b

Explanation: Flow chart shows basic blocks.

5. _____ or scanning is the process where the stream of characters making up the source program is read from left to right and grouped into tokens.

- a) Lexical Analysis**
- b) Diversion**
- c) Modelling**
- d) None of the mentioned**

Answer: a

Explanation: Lexical analysis is the process of converting a sequence of characters into a sequence of tokens.

6. _____ is a graph representation of a derivation.

- a) The parse tree**
- b) Oct tree**
- c) Binary tree**
- d) None of the mentioned**

Answer: a

Explanation: Parse tree is a representation of the derivation.

7. Which of the following symbols table implementation is based on the property of locality of reference?

- a) Hash Table**
- b) Search tree**
- c) Self organizing list**
- d) Linear list**

Answer: c

Explanation: Self Organizing list is based on locality of reference.

8. Assume that the SLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. Hence which one is true?

- a) N_1 is necessarily less than n_2
- b) N_1 is necessarily equal to n_2
- c) N_1 is necessarily greater than n_2
- d) None of the mentioned

Answer: b

Explanation: The output of lexical analyzer is output token.

1. Which of these is also known as look-head LR parser?

- a) SLR
- b) LR
- c) LLR
- d) None of the mentioned

Answer: c

Explanation: LLR is the look ahead parser.

2. What is the similarity between LR, LALR and SLR?

- a) Use same algorithm, but different parsing table
- b) Same parsing table, but different algorithm
- c) Their Parsing tables and algorithm are similar but uses top down approach
- d) Both Parsing tables and algorithm are different

Answer: a

Explanation: The common grounds of these 3 parser is the algorithm but parsing table is different.

3. An LR-parser can detect a syntactic error as soon as _____

- a) The parsing starts
- b) It is possible to do so a left-to-right scan of the input
- c) It is possible to do so a right-to-left scan of the input
- d) Parsing ends

Answer: b

Explanation: Error is found when it the input string is scanned.

4. Which of these is true about LR parsing?

- a) Is most general non-backtracking shift-reduce parsing
- b) It is still efficient
- c) Is most general non-backtracking shift-reduce parsing & It is still efficient
- d) None of the mentioned

Answer: c

Explanation: LR parsers are a type of bottom-up parsers that efficiently handle deterministic context-free languages in guaranteed linear time.

5. Which of the following is incorrect for the actions of A LR-Parser i) shift s ii) reduce A->B iii) Accept iv) reject?

- a) Only i)
- b) i) and ii)
- c) i), ii) and iii)
- d) i), ii) , iii) and iv)

Answer: c

Explanation: Only reject out of the following is a correct LR parser action.

6. If a state does not know whether it will make a shift operation or reduction for a terminal is called _____

- a) Shift/reduce conflict

- b) Reduce /shift conflict
- c) Shift conflict
- d) Reduce conflict

Answer: a

Explanation: As the name suggests that the conflict is between shift and reduce hence it is called shift reduce conflict.

7. When there is a reduce/reduce conflict?

- a) If a state does not know whether it will make a shift operation using the production rule i or j for a terminal
- b) If a state does not know whether it will make a shift or reduction operation using the production rule i or j for a terminal
- c) If a state does not know whether it will make a reduction operation using the production rule i or j for a terminal
- d) None of the mentioned

Answer: c

Explanation: It occurs when If a state does not know whether it will make a reduction operation using the production rule i or j for a terminal.

8. When β (in the LR(1) item $A \rightarrow \beta.a, a$) is not empty, the look-head _____

- a) Will be affecting
- b) Does not have any affect
- c) Shift will take place
- d) Reduction will take place

Answer: b

Explanation: There is no terminal before the non terminal beta.

9. When β is empty ($A \rightarrow \beta., a$), the reduction by $A \rightarrow a$ is done by _____

- a) If next symbol is a terminal
- b) Only If the next input symbol is a
- c) Only If the next input symbol is A
- d) Only if the next input symbol is a

Answer: d

Explanation: The next token is considered in this case it's a.

10. The construction of the canonical collection of the sets of LR (1) items are similar to the construction of the canonical collection of the sets of LR (0) items. Which is an exception?

- a) Closure and goto operations work a little bit different
- b) Closure and goto operations work similarly
- c) Closure and additive operations work a little bit different
- d) Closure and associatively operations work a little bit different

Answer: a

Explanation: Closure and goto do work differently in case of LR (0) and LR (1).

1. What is terminal table?

- a) Contains all constants in the program
- b) Is a permanent table of decision rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure
- c) Consist of a full or partial list of the token is as they appear in the program created by lexical analysis and used for syntax analysis and interpretation
- d) Is a permanent table which lists all keywords and special symbols of the language in symbolic form

Answer: d

Explanation: A permanent database that has entry for each terminal symbols such as arithmetic operators, keywords, punctuation characters such as ';;', ',' etc Fields: Name of the symbol.

2. Advantage of incorporating the macro-processor into pass 1 is that _____
- a) Many functions have to be implemented twice
 - b) Functions are combined not necessarily creating intermediate files as output from the macro-processor and input to the assembler
 - c) More flexibility is provided to the programmer in that he may use all the features of the assembler in conjunction with macros
 - d) All of the mentioned

Answer: d

Explanation: A general-purpose macro processor or general purpose pre-processor is a macro designed primarily for string manipulation, macro definition.

3. Which of the following is a phase of a compilation process?
- a) Lexical Analysis
 - b) Code Generation
 - c) Lexical Analysis & Code Generation
 - d) None of the mentioned

Answer: c

Explanation: Lexical analysis and code generation is a phase of compilation process.

4. System program such as compiler are designed so that they are _____
- a) Re-enterable
 - b) Non reusable
 - c) Serially usable
 - d) None of the mentioned

Answer: a

Explanation: Re-enterable is the keyword for compiler being designed.

5. A series of statements explaining how the data is to be processed is called _____
- a) Assembly
 - b) Machine
 - c) COBOL
 - d) Program

Answer: d

Explanation: A program is a sequence of instructions, written to perform a task by computer. It requires programs to function, typically executing the program's instructions in a central processor.

6. A loader is a program that _____
- a) Program that places functions into memory and prepares them for execution
 - b) Program that automates the translation of assembly language into machine language
 - c) Program accepting another program written in a high level language and produces as object program
 - d) None of the mentioned

Answer: a

Explanation: A loader is the part of an operating system that is responsible for loading programs and libraries. It is important in the process of placing the programs into memory and executing them.

7. A system program that setup an executable program in main memory ready for execution is?
- a) Assembler
 - b) Linker
 - c) Loader
 - d) Load and go

Answer: c

Explanation: A loader is the part of an operating system that is responsible for loading programs and libraries. It is one

of the essential stages in the process of starting a program, as it places programs into memory and prepares them for execution.

8. Which of the following system program forgoes the production of object code to generate absolute machine code and load it into the physical main storage location from which it will be executed immediately upon completion of the assembly?

- a) Two pass assembler**
- b) Load and go assembler**
- c) Macro processor**
- d) Linker**

Answer: b

Explanation: A load and go assembler generates absolute machine code and loads it to physical memory.

9. Uniform symbol table _____

- a) Has all constants in the program**
- b) Permanent table of rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure**
- c) Consists of full or partial list of the tokens as they appear in the program created by Lexical analysis and used for syntax analysis and interpretation**
- d) A permanent table which has all key words and special symbols of the language in symbolic form**

Answer: c

Explanation: Each pass scans the program, the first pass generates the symbol table and the second pass generates the machine code.

10. Assembler is a program that _____

- a) Puts programs into memory and executes them**
- b) Translates the assembly language into machine language**
- c) Writes in high level language and produces an object program**
- d) None of the mentioned**

Answer: b

Explanation: An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations.

1. Compiler can diagnose _____

- a) Grammatical errors only**
- b) Logical errors only**
- c) Grammatical and logical errors**
- d) None of the mentioned**

Answer: a

Explanation: Only syntactical errors can be detected by the compiler.

2. A simple two-pass assembler does which of the following in the first pass?

- a) It allocates space for the literals**
- b) Calculates total length of the program**
- c) Symbol table is built for the symbols and their value**
- d) All of the mentioned**

Answer: d

Explanation: A two-pass assembler. Each pass scans the program, the first pass generates the symbol table and the second pass generates the machine code.

3. A system program that set-up an executable program in the main memory ready for execution is?

- a) Assembler**
- b) Linker**

c) Loader

d) Text editor

Answer: c

Explanation: A loader is the part of an operating system that is responsible for loading programs and libraries. It is important that with the starting of a program, as it places programs into memory and executes it.

4. A compiler is a program that _____

a) Program is put into memory and executes it

b) Translation of assembly language into machine language

c) Acceptance of a program written in a high level language and produces an object program

d) None of the mentioned

Answer: c

Explanation: A compiler is a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code).

5. A programmer by mistake writes multiplication instead of division, such error can be detected by?

a) Compiler

b) Interpreter

c) Compiler or interpreter test

d) None of the mentioned

Answer: d

Explanation: No Logical errors can be detected.

6. The computer language generally translated to pseudocode is _____

a) Assembly

b) Machine

c) Pascal

d) FORTRAN

Answer: a

Explanation: An assembly language (or assembler language) is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions.

7. A system program that combines separately compiled modules of a program into a form suitable for execution is?

a) Assembler

b) Linking Loader

c) Cross Compiler

d) None of the mentioned

Answer: b

Explanation: A loader which combines the functions of a relocating loader with the ability to combine a number of program segments that have been independently compiled into an executable program.

8. In which way a macro processor for assembly language can be implemented?

a) Independent two-pass processor

b) Independent one-pass processor

c) Processor put into pass 1 of a standard two pass assembler

d) All of the mentioned

Answer: d

Explanation: A general-purpose macro processor or general purpose preprocessor is a macro designed for string manipulation, macro definition.

9. Resolution of externally defined symbols is performed by _____

- a) Linker
- b) Loader
- c) Compiler
- d) Interpreter

Answer: a

Explanation: A linker or link editor is a computer program that takes one or more object files generated by a compiler and combines them into a single executable file, library file, or another object file.

10. A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar $S \rightarrow xxW$ (PRINT "1") $S \rightarrow y \{ \text{print " 2 " } \}$ $S \rightarrow Sz \{ \text{print " 3 " } \}$) What is the translation of xxxxyzz using the syntax directed translation scheme described by the above rules?

- a) 23131
- b) 11233
- c) 11231
- d) 33211

Answer: a

Explanation: Initially 2 is printed then 3 then 1 3 1.

1. In operator precedence parsing whose precedence relations are defined _____

- a) For all pair of non-terminals
- b) For all pair of terminals
- c) To delimit the handle
- d) None of the mentioned

Answer: a

Explanation: There are two important properties for these operator precedence parsers is that it does not appear on the right side of any production and no production has two adjacent no terminals.

2. LR parsers are attractive because _____

- a) It can be constructed to recognize CFG corresponding to almost all programming constructs
- b) It does not backtrack
- c) It can be constructed to recognize CFG corresponding to almost all programming constructs & It does not backtrack
- d) None of the mentioned

Answer: c

Explanation: These above mentioned are the reasons why LR parser is considered to be attractive

3. Which is the most powerful parser?

- a) SLR
- b) LALR
- c) Canonical LR
- d) Operator Precedence

Answer: c

Explanation: The most powerful parser is Canonical LR

4. $(Z, *)$ be a structure, and $*$ is defined by $n * m = \text{maximum}(n, m)$ Which of the following statements is true for $(Z, *)$?

- a) $(Z, *)$ is a monoid
- b) $(Z, *)$ is an algebraic group
- c) $(Z, *)$ is a group
- d) None of the mentioned

Answer: d

Explanation: It is neither a monoid nor a simple group nor algebraic group

5. The address code involves _____

- a) Exactly 3 address**
- b) At most Three address**
- c) No unary operators**
- d) None of the mentioned**

Answer: d

Explanation: In computer science, three-address is an intermediate code used by optimizing compilers to aid in the implementation of code-improving transformations.

6. An intermediate code form is _____

- a) Postfix Notation**
- b) Syntax Trees**
- c) Three address code**
- d) All of the mentioned**

Answer: d

Explanation: Intermediate code generator takes input from its predecessor phase, semantic analyzer, in the form of an annotated syntax tree.

1. In a single pass assembler, most of the forward references can be avoided by putting the restriction _____

- a) On the number of strings/life reacts**
- b) Code segment to be defined after data segment**
- c) On unconditional rump**
- d) None of the mentioned**

Answer: b

Explanation: A single pass assembler scans the program only once and creates the equivalent binary program.

2. The method which merges the bodies of two loops is?

- a) Loop rolling**
- b) Loop jamming**
- c) Constant folding**
- d) None of the mentioned**

Answer: b

Explanation: In computer science, loop fusion (or loop jamming) is a compiler optimization and loop transformation which replaces multiple loops with a single one.

3. Assembly code data base is associated with _____

- a) Code is converted into assembly**
- b) Table of rules in the form of patterns for matching with the uniform symbol table to discover syntactic structure**
- c) All of the mentioned**
- d) None of the mentioned**

Answer: a

Explanation: An assembly language is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions.

4. The process manager has to keep track of _____

- a) Status of each program**
- b) Information to a programmer using the system**
- c) Status of each program & Information to a programmer using the system**
- d) None of the mentioned**

Answer: c

Explanation: Process manager keep track of the status and info about the program.

5. What is the function of the syntax phase?

- a) recognize the language and to cal the appropriate action routines that will generate the intermediate form or matrix for these constructs
- b) Build a literal table and an identifier table
- c) Build a uniform symbol table
- d) Parse the source program into the basic elements or tokens of the language

Answer: a

Explanation: In this phase symbol table is created by the compiler which contains the list of lexemes or tokens.

6. If E be a shifting operation applied to a function f, such that $E(f) = f(x + \beta)$, then?

- a) $E(\alpha f + \beta g) = \alpha E(f) + \beta E(g)$
- b) $E(\alpha f + \beta g) = (\alpha + \beta) + E(f + g)$
- c) $E(\alpha f + \beta g) = \alpha E(f + g\beta)$
- d) $E(\alpha f + \beta g) = \alpha\beta E(f + g)$

Answer: a

Explanation: Shifting operation when performed gives this result.

7. Pass I _____

- a) Assign address to all statements
- b) Save the values assigned to all labels for use in pass 2
- c) Perform some processing
- d) All of the mentioned

Answer: d

Explanation: The pass 1 of a compiler the above mentioned functions are performed

8. Which table is a permanent database that has an entry for each terminal symbol?

- a) Terminal Table
- b) Literal Table
- c) Identifier Table
- d) None of the mentioned

Answer: a

Explanation: A database that has entry for each terminal symbols such as arithmetic operators, keywords, punctuation characters such as ';', ',', etc Fields: Name of the symbol.

9. Which of the following functions is performed by loader?

- a) Allocate memory for the programs and resolve symbolic references between objects decks
- b) Address dependent locations, such as address constants, to correspond to the allocated space
- c) Physically place the machine instructions and data into memory
- d) All of the mentioned

Answer: d

Explanation: A loader is the part of an operating system that is responsible for loading programs and libraries.

10. The root directory of a disk should be placed _____

- a) At a fixed address in main memory
- b) At a fixed location on the disk
- c) Anywhere on the disk
- d) None of the mentioned

Answer: b

Explanation: Root directory is placed at a fixed disk location

1. The segment base is specified using the register named is?

- a) ORG instructions**
- b) TITLE instruction**
- c) ASSUME instruction**
- d) SEGMENT instruction**

Answer: a

Explanation: ORG segment base register is used to specify the register.

2. In what module multiple instances of execution will yield the same result even if one instance has not terminated before the next one has begun?

- a) Non usable module**
- b) Serially usable**
- c) Re-enter-able**
- d) None of the mentioned**

Answer: c

Explanation: Re enter-able module is the reason why the compiler is used in the first place.

3. Dividing a project into segments and smaller units in order to simplify design and programming efforts is called?

- a) Modular approach**
- b) Top down approach**
- c) Bottom up approach**
- d) Left right approach**

Answer: a

Explanation: Modular design, or “modularity in design”, is a design approach that subdivides a system into smaller parts called modules or skids that can be independently created and then used in different systems.

4. Which one of the following is the tightest upper bound that represents the time complexity of inserting an object into a binary search tree of n nodes?

- a) O(1)**
- b) O(long)**
- c) O(n)**
- d) O(long)**

Answer: c

Explanation: For skewed binary search tree on n nodes, the upper bound to insert a node is O (n).

5. Which of the following is true for machine language?

- a) Continuous execution of program segments**
- b) Depicting flow of data in a system**
- c) A sequence of instructions which solves a problem**
- d) The language which interacts with the computer using only the binary digits 1 and 0**

Answer: d

Explanation: Machine code or machine language is a set of instructions executed directly by a computer’s central processing unit (CPU). Each instruction performs a very specific task.

6. Software that measures, monitors, and controls events are _____

- a) System S/w**
- b) Real Time software**
- c) Scientific software**
- d) Business Software**

Answer: b

Explanation: In computer science, real-time computing (RTC), or reactive computing describes hardware and software systems subject to a “real-time constraint”, for example operational deadlines from event to system response.

7. A linker is given object module for a set of programs that were compiled separately. What is not true about an object module?
- a) Object code
 - b) Relocation bits
 - c) Names and locations of all external symbols defined in the object module
 - d) Absolute addresses of internal symbols

Answer: d

Explanation: A linker or link editor is a computer program that takes one or more object files generated by a compiler and combines them into a single executable file, library file, or another object file.

8. The table created by lexical analysis to describe all literals used in the source program is?
- a) Terminal table
 - b) Literal table
 - c) Identifier table
 - d) Reductions

Answer: b

Explanation: In computer science, and specifically in compiler and assembler design, literal pool is a lookup table used to hold literals during assembly and execution.

9. Which loader function is accomplished by loader?
- a) Reallocation
 - b) Allocation
 - c) Linking
 - d) Loading

Answer: d

Explanation: Function of a loader.

10. What is Pass 2?
- a) Assemble instruction and generate data
 - b) Perform processing of assembler
 - c) Write the object program
 - d) All of the mentioned

Answer: d

Explanation: A multi-pass compiler is a type of compiler that processes the source code or abstract syntax tree of a program several times. Each pass takes the result of the previous pass as the input and creates an intermediate output.

1. Which is not true about syntax and semantic parts of a computer language?
- a) Semantics is checked mechanically by a computer
 - b) Semantics is the responsibility of the programmer
 - c) All of the mentioned
 - d) None of the mentioned

Answer: d

Explanation: Both in terms of semantics is not true.

2. Which of the following statement is true?
- a) SLR powerful than LALR
 - b) LALR powerful than Canonical LR parser
 - c) Canonical LR powerful than LALR parser
 - d) The parsers SLR= Canonical LR=LALR

Answer: c

Explanation: $LR > LALR > SLR$ In terms of the parser.

3. Which of the following features cannot be captured by CFG?

- a) Syntax of if-then-else statements**
- b) Syntax of recursive procedures**
- c) A variable is declared before its use**
- d) Matching nested parenthesis**

Answer: d

Explanation: It is because, it is equivalent to recognizing us, where the first w is the declaration and the second is its use, we are not a CFG.

4. In which of the following no information hiding is done?

- a) Compile prig 1, prig 2**
- b) Run test, prog**
- c) Load R1, A**
- d) 001001000010101**

Answer: d

Explanation: The entire binary symbol gives information.

5. The identification of common sub-expression and replacement of run-time computations by compile-time computations is _____

- a) Local optimization**
- b) Loop optimization**
- c) Constant folding**
- d) Data flow analysis**

Answer: c

Explanation: Constant folding is the process of recognizing and evaluating constant expressions at compile time rather than computing them at runtime. Terms in constant expressions are typically simple literals they may also be variables whose values are assigned at compile time.

6. The graph that shows basic blocks and their successor relationship is called _____

- a) Dag**
- b) Flow Graph**
- c) Control Graph**
- d) Hamilton Graph**

Answer: b

Explanation: Flow graph shows the basic blocks.

7. The specific task storage manager performs _____

- a) Allocation/ deal location of programs**
- b) Protection of storage area assigned to the program**
- c) Allocation/ deal location of programs & protection of storage area assigned to the program**
- d) None of the mentioned**

Answer: c

Explanation: Its basic function is that of the task storage manager.

8. When a computer is rebooted, a special type of loader is executed called?

- a) Compile and GO ” loader**
- b) Boot loader**
- c) Bootstrap Loader**
- d) Relating Loader**

Answer: c

Explanation: A boot loader, is a small program that places the operating system (OS) of a computer into memory.

9. What is the disadvantage of "Compile and GO" loading scheme?

- a) Memory is wasted because the case occupied by the assembler is unavailable to the object program**
- b) Necessary to translate the users program**
- c) It is very difficult to handle multiple segments, even when the source programs are in different languages and to produce orderly modular programs**
- d) All of the mentioned**

Answer: d

Explanation: In computer programming, a compile and go system, compile, load, and go system, assemble and go system, or load and go system[1][2][3] is a programming language processor in which the compilation, assembly, or link steps are not separated from program execution.

10. What is the function of the storage assignment?

- a) Assign storage to all variables referenced in the source program**
- b) Assign storage to all temporary locations that are necessary for intermediate results**
- c) Assign storage to literals, and to ensure that the storage is allocated and appropriate locations are initialized**
- d) All of the mentioned**

Answer: d

Explanation: The storage assignment performs the above mentioned tasks.

1. A non relocatable program is the one which?

- a) Cannot execute in any area of storage other than the one designated**
- b) Consists of a program and information for its relocation**
- c) None of the mentioned**
- d) All of the mentioned**

Answer: a

Explanation: A non reloadable program is one which cannot be made to execute in any area of storage other than the one designated for it at the time of its coding or translation.

2. A relocatable program form is one which?

- a) Cannot execute in any area of storage other than the one designated**
- b) Consists of a program and information for its relocation**
- c) None of the mentioned**
- d) All of the mentioned**

Answer: c

Explanation: A relocatable program form is one which consists of a program and relevant information for its relocation. Using this information it is possible to relocate the program to execute from a storage area then the one designated for it at the time of its coding or translation.

3. A self-relocating program is one which?

- a) Cannot execute in any area of storage other than the one designated**
- b) Consists of a program and information for its relocation**
- c) None of the mentioned**
- d) All of the mentioned**

Answer: c

Explanation: A self-relocating program is a program which can perform the relocation itself

- A table of information about address sensitive instruction in the program.*
- Relocating logic that can perform the relocation of the address sensitive instructions.*

4. Scissoring enables _____

- a) A part of data to be displayed**
- b) Entire data to be displayed**
- c) Full data display on full screen**
- d) No data to be displayed**

Answer: a

Explanation: It displays a part of the data.

5. Which of the following can be accessed by the transfer vector approach of linking?

- a) External data segments**
- b) External sub-routines**
- c) Data located in other procedure**
- d) All of the mentioned**

Answer: b

Explanation: External subroutines are routines that are created and maintained separately from the program that will be calling them.

6. Relocation bits used by relocating loader are specified by _____

- a) Relocating loader itself**
- b) Linker**
- c) Assembler**
- d) Macro processor**

Answer: b

Explanation: A linker or link editor is a computer program that takes one or more object files generated by a compiler and combines them into a single executable file, library file, or another object file.

7. Generation of intermediate code based on a abstract machine model is useful in compilers because _____

- a) Implementation of lexical analysis and syntax analysis is made easier**
- b) Writing for intermediate code generation**
- c) Portability of the front end of the compiler**
- d) None of the mentioned**

Answer: a

Explanation: Intermediate code generator receives input from its predecessor phase, semantic analyzer, in the form of an annotated syntax tree.

8. Which of the following module does not incorporate initialization of values changed by the module?

- a) Non reusable module**
- b) Serially reusable module**
- c) Re-enterable module**
- d) All of the mentioned**

Answer: a

Explanation: Non reusable models can be used once for a purpose they can't be modified and used again.

9. The best way to compare the different implementations of symbol table is to compare the time required to _____

- a) Add a new name**
- b) Make an enquiry**
- c) Add a new name and make an enquiry**
- d) All of the mentioned**

Answer: d

Explanation: These are the different implementations of the symbol table as mentioned above.

1. Dynamic linking can cause security concerns because _____

- a) Security is dynamic**
- b) None of the mentioned**
- c) Security is dynamic & Cryptographic procedures are not available for dynamic linking**
- d) Cryptographic procedures are not available for dynamic linking**

Answer: d

Explanation: Required by the program are linked during run time.

2. Which of the following statements is FALSE?

- a) In up-typed languages, values do not have any types & Dynamic languages have no variables types
- b) In up-typed languages, values do not have any types
- c) Dynamic languages have no variables types
- d) None of the mentioned

Answer: c

Explanation: They have dynamically changing types.

- a) LL (1)
- b) SLR (1) but not LL (1)
- c) LALR (1) but not SLR (1)
- d) LR (1) but not LALR (1)

Answer: c

Explanation: The grammar is CLR and can be reduced to LALR.

4. The 3-address code sequence generated by which definition?

```
S -> id: = E "  
newtemp ();  
  gen(t . place . place;); .place t} "  
.place .place;}  
For the statement 'X: = Y + Z ',
```

- a) $X = Y + Z$
- b) $t1 = Y + Z; X = t1$
- c) $t1 = Y; t2 = t1 + Z; X = t2$
- d) $t1 = Y; t2 = Z; t3 = t2; X = t3$

Answer: d

Explanation: In 3-address code we use temporary variables.

5. If the programming language uses static scoping and call by need parameter passing mechanism, the values printed will be?

- a) 115, 220
- b) 25, 220
- c) 25, 15
- d) 115, 105

Answer: d

Explanation: So $i = 100$ & $j = 5$ $P(i + j) = P(100 + 5) = P(105)$ So $x = 105$ $x + 10 = 105 + 10 = 115$ So 115 & 105 will be printed.

6. If dynamic scoping is used and call by name mechanism is used, the values printed will be?

- a) 115, 220
- b) 25, 220
- c) 25, 15
- d) 115, 105

Answer: b

Explanation: Since $x = i + j$ & in $P(x)$ $i = 200$ & $j = 20$ $x = 200 + 20 = 220$ & printing $(x + 10)$ $9. = i + j + 10 = 10 + 5 + 10 = 25$.

7. What will be the output of the following code?

```
P x =new Q();
```

```
Q y =new Q();
P z =new Q();
x. f (1);
((P) y). f (1);
z.f(1);
```

- a) 1 2 1
- b) 2 1 1
- c) 2 1 2
- d) 2 2 2

Answer:d
Explanation: Here,

```
1. Px = newQ();
2. Qy = newQ();
3. Pz = newQ();
4. x : f(1);
print 2 # i = 2
5. ((P) y) :f(1);
6. z : f(1) print 2 # i = 2
```

but line 6. will print 2 because typecast to parent class can't prevent over ridding.

8. Which of the following is NOT an advantage of using shared, dynamically linked libraries as compared to statically linked libraries?

- a) Smaller sizes of executable
- b) Lesser overall page fault rate in the system
- c) Faster program start-up
- d) Existing programs need not be re-linked to take advantage of newer versions of libraries

Answer: c
Explanation: No need for re-linking if newer versions of libraries are there.

9. Which grammar violate rules of an operator grammar?

```
(i)      P-> QR
(ii)     P -> Q s R
(iii)    P -> ε
(iv)     P ->Q t
```

- a) (i) only
- b) (i) and (iii) only
- c) (ii) and (iii) only
- d) (iii) and (iv) only

Answer: b
Explanation: (I) P " QR is not possible since two NT should include one operator as Terminal.

10. Consider the grammar rule. E -> E1 – E2 for arithmetic expressions. If E1 and E2 do not have any common sub expression, in order to get the shortest possible code.

- a) E1 should be evaluated first
- b) E2 should be evaluated first
- c) Evaluation of E1 and E2 should necessarily be interleaved
- d) Order of evaluation of E1 and E2 is of no consequence

Answer: b
Explanation: E1 is to be kept in accumulator & accumulator is required for operations to evaluate E2.

1. The grammar A ->AA |(A)| ε is not suitable for predictive-parsing because the grammar is _____

- a) Ambiguous

- b) Left-recursive
- c) Right-recursive
- d) An operator-grammar

Answer: a

Explanation: The grammar is ambiguous.

- For a sentence $n + n$, form of the reduction are _____
- a) $n, E + n$ and $E + n \# n$
 - b) $n, E + n$ and $E + E \# n$
 - c) $n, n + n$ and $n + n \# n$
 - d) $n, E + n$ and $E \# n$

Answer: d

Explanation: During reduction the order is reverse. So $\{E \rightarrow n, E \rightarrow E + n, E \rightarrow E \# n\}$.Answer: b

Explanation: The no. of states for $SLR(1)$ & $LALR(1)$ are equal so $n \neq n^3$, but $CLR(1)$ or $LR(1)$ will have no. of states greater than $LALR$ & $LR(0)$ both.

4. Consider line 3. Identify the compiler’s response about this line while creating the object-module

```
Int main()
{ int I,N;
  Fro (I,I N,I);
}
```

- a) No compilation error
- b) Only a lexical error
- c) Only syntactic errors
- d) Both lexical and syntactic errors

Answer: c

Explanation: There are no lexical errors for C because all the wrong spelled keywords.

5. Which one of the following is true about the action of yacc for the given grammar?
- a) It detects recursion and eliminates recursion
 - b) It detects reduce-reduce conflict and resolves
 - c) It detects shift-reduce conflict and resolves the conflict in favor of a shift over a reduce action
 - d) It detects shift-reduce conflict and resolves the conflict in favor of a reduce over a shift action

Answer: c

Explanation: Yacc tool is used to create a $LALR(1)$ parser.

6. What precedence and associativity properties does the generated parser realize?
- a) Equal precedence and left associativity; expression is evaluated to 7
 - b) Equal precedence and right associativity, expression is evaluated to 9
 - c) Precedence of ‘x’ is higher than that of ‘+’, and both operators are left associative; expression is evaluated to 7
 - d) Precedence of ‘ # ‘ is higher than that of ‘#’, and both operators are left associative; expression is evaluated to 9

Answer: b

Explanation: The grammar has equal precedence and it is also ambiguous.

7. Consider the following grammar:

```
S > S * E
S > E
E > F + E
E > F
F > id
```

Consider the following LR(0) items corresponding to the grammar above.


```
(i)      S -> S * .E
(ii)     E -> F . + E
(iii)    E -> F + .E
```

Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?

- a) (i) and (ii)
- b) (ii) and (iii)
- c) (i) and (iii)
- d) None of the mentioned

Answer: c

Explanation: If $S \rightarrow S$: E is in $LR(0)$ then $E \rightarrow F + : E$ will also be there because both of them has ' : ' before E .

8. Consider the following grammar:

```
S > FR
R > * S | ε
F > id
```

In the predictive parser table, M , of the grammar the entries $M[S, id]$ and $M[R, \$]$ respectively.

- a) $\{S \rightarrow FR\}$ and $\{R \rightarrow \epsilon\}$
- b) $\{S \rightarrow FR\}$ and $\{\}$
- c) $\{S \rightarrow FR\}$ and $\{R \rightarrow * S\}$
- d) $\{F \rightarrow id\}$ and $\{R \rightarrow \epsilon\}$

Answer: a

Explanation: The predictive parser table is given as. Non Terminal) id \$ $S \rightarrow FR$

$F \rightarrow id$ $R \rightarrow$

$S \rightarrow$

$R \rightarrow ! R \rightarrow !$

So at $M[S, id] = \{S \rightarrow FR\}$ $M[R, \$] = \{R \rightarrow !\}$

9. Consider the following translation scheme.

```
S > ER
R > * E{print(' * ')};
R | f
E > F + E{print(' + ')}; | F F > (S) | id{print(id.value);}
```

Here id is a taken that represents an integer and $id.value$ represents the corresponding integer value. For an input ' $2 * 3 + 4$ ', this translation scheme prints?

- a) $2 * 3 + 4$
- b) $2 * + 3 4$
- c) $2 3 * 4 +$
- d) $2 3 4 + *$

Answer: b

Explanation: Input string $2 * 3 + 4$ $S \rightarrow ER$ FR $idR \{print(2)\} id)ER \{print()\} id) F+ER \{print(+)\} id) id + ER \{print(3)\} id) id) id + id$ So $2 * 3 + 4$ are printed.

Which one to the following false?

- a) The code contains loop-in variant computation
- b) There is scope of common sub-expression elimination in this code
- c) There is scope strength reduction in this code
- d) There is scope of dead code elimination in this code

Answer: d

Explanation: All the statements are true except option last since there is no dead code to get eliminated.

1. The below grammar and the semantic rules are fed to a yacc tool (which is an LALR (1) parser generator) for

parsing and evaluating arithmetic expressions. Which one of the following is true about the action of yacc for the given grammar?

E -> number	Eval	number val	
E	E .val	E .VAL	E .val
E # E	E .val	E .VAL	E .val
;			

- a) It detects recursion and eliminates recursion
- b) It detects reduce-reduce conflict and resolves
- c) It detects shift-reduce conflict and resolves the conflict in favor of a shift over a reduce action
- d) It detects shift-reduce conflict and resolves the conflict in favor of a reduce over a shift action

Answer: c

Explanation: Yacc tool is used to create a LALR (1) parser. This parser can detect the conflicts but to resolve the conflicts it actually prefers shift over reduce action.

2. Assume the conflicts part (a) of this question are resolved and an LALR (1) parser is generated for parsing arithmetic expressions as per the given grammar. Consider an expression 3 # 2 + 1. What precedence and associativity properties does the generated parser realize?

E -> number	Eval	number val	
E	E .val	E .VAL	E .val
E # E	E .val	E .VAL	E .val
;			

- a) Equal precedence and left associativity; expression is evaluated to 7
- b) Equal precedence and right associativity, expression is evaluated to 9
- c) Precedence of ‘x’ is higher than that of ‘+’, and both operators are left associative; expression is evaluated to 7
- d) Precedence of ‘ # ‘ is higher than that of ‘#’, and both operators are left associative; expression is evaluated to 9

Answer: b

Explanation: The grammar has equal precedence and it is also ambiguous. Since LALR (1) parser prefer shift over reduce so + operation will be executed here before). $2 + 1 = 3$ & $3 \# 3 = 9$ also the operators are right associative.

3. Consider the following grammar:

S -> S * E
S -> E
E -> F + E
E -> F
F -> id

Consider the following LR (0) items corresponding to the grammar above.

(i) S -> S * .E
(ii) E -> F. + E
(iii) E "F + .E

- Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?
- a) (ii)
 - b) (i) and (iii)
 - c) (iii)
 - d) None of the mentioned

Answer: C

Explanation: If $S \rightarrow S$: E is in LR (0) then $E \rightarrow F +$: E will also be there because both of them has ‘: ‘ before E.

4. Consider the following grammar:

S -> FR
R -> * S ε

F -> id

In the predictive parser table, M, of the grammar the entries M [S, id] and M [R, \$] respectively.

- a) {S " FR} and {R " ε}
- b) {S " FR} and {}
- c) {S " FR} and {R " * S}
- d) {F " id} and {R " ε}

Answer: a

Explanation: The predictive parser table is given as. Non Terminal) id \$ S S "FR F F "id R R ") S R " ! R " ! So at M [S, id] = { S " FR} M [R,\$] = {R " !}.

5. Consider the following translation scheme.

```
S -> ER
R -> * E{print(' * ');
R | f
E -> F + E{print(' + '); | F
F -> (S) | id{print(id.value);}
```

Here id is a taken that represents an integer and id. value represents the corresponding integer value. For an input ‘2 * 3 + 4’, this translation scheme prints?

- a) 2 * 3 + 4
- b) 2 * + 3 4
- c) 2 3 * 4 +
- d) 2 3 4 + *

Answer: b

Explanation: Input string 2) 3 + 4 S " ER FR idR {print(2)} id)ER {print())} id) F+ER {print(+)}id) id + ER {print(3)} id) id) id +id So 2)+ 3 4 are printed.

Which one to the following false?

- a) The code contains loop-in variant computation
- b) There is scope of common sub-expression elimination in this code
- c) There is scope strength reduction in this code
- d) There is scope of dead code elimination in this code

Answer: d

Explanation: All the statements are true except option (There is scope of dead code elimination in this code) since there is no dead code to get eliminated.

b)

S ->AC | CB

c)

```
S -> aS | Sb | a | b
C -> aCb | a | b
A -> aA | ε
B -> Bb | ε
```

d)

S -> ACCB

8. In the correct grammar above, what is the length of the derivation (number of steps starting from S to generate the string a l b m with l ! m?

- a) max (l, m) + 2
- b) l+m+2
- c) l + m + 3

d) $\max(l, m) + 3$

Answer: a

Explanation: It is very clear from the previous solution that the no. of steps required depend upon the no. of a's & b's which ever is higher & exceeds by 2 due to $S \rightarrow AC \mid CB$ & $C \rightarrow !$ So $\max(l, m) + 2$.

9. Which one of the following is a top-down parser?

- a) Recursive descent parser**
- b) Operator precedence parser**
- c) An LR(k) parser**
- d) An LALR(k) parser**

Answer: a

Explanation: Clearly LR & LALR are not top down they are bottom up parsers. Also not operator precedence parser: it yes recursive descent parser is top down parser. Starts from start symbol & derives the terminal string.

10. Consider the grammar with non-terminals.

```
S -> AC | CB
C -> aCb | !
C -> aCb | !
A -> aA | !
A -> aA | a
B -> Bb | !
B -> bB | b
```

The grammar is NOTLL(1) because _____

- a) It is left recursive**
- b) It is right recursive**
- c) It is ambiguous**
- d) It is not context-free**

Answer: a

Explanation: The grammar has production $S \rightarrow iCtSSl$ here the right hand side of grammar has the same symbol as left side. So the grammar is left recursive. The grammar is not ambiguous. Answer: a

Explanation: LL(1) parsers can recognize the regular grammars also LL(1) is subset of LR(1) or CLR grammar so it also recognizes regular sets. So both accept regular grammar.

2. In a simplified computer the instructions are:

```
OP Rj, Ri - Performs Rj OP Ri and stores the result in register Ri
OP m, Ri - Performs val OP Ri and stores the result in Ri. value denotes the content of memory location m.
MCVm, Ri - Moves the content of memory location m to register Ri.
MCVm, Ri, m - Moves the content of register Ri to memory location m.
```

The computer has only two registers, and OP is either ADD or SUB. Consider the following basic block:

```
t1 = a + b
t2 = c + d
t3 = e - t2
t4 = t1 - t2
```

Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?

- a) 2**
- b) 3**
- c) 5**
- d) 6**

Answer: b

Explanation: The operation sequence would be

MOV a, R1
 ADD b, R1 {R 1 = t1
 MOV c, R2
 ADD d, R2 {R 2 = t2
 SUB e, R2 {t 3 = e - R 2 = R2
 SUB R 1, R2 {R 2 = t4
 MOV R 2, t4 {finally in memory
 Totally no. of move operation is 3.

3. Which of the following strings is generated by the grammar?

S->bA	S->aB
A->a	B->b
A->aS	B->bS
A->bAA	B->aBB

- a) aaaabb
- b) aabbbb
- c) aabbab
- d) abbbba

Answer: c
 Explanation: aabbab S " aB " aaBB " aabSB " aabbAB " aabbab

4. How many derivation trees are there?

S->bA	S->aB
A->a	B->b
A->aS	B->bS
A->bAA	B->aBB

- a) 1
- b) 2
- c) 3
- d) 4

Answer: b
 Explanation: For the derivation two trees are possible So due to ambiguity 2 trees are possible.

5. Which of the following describes a handle (as applicable to LR-parsing) appropriately?
- a) It is the position in a sentential form where the next shift or reduce operation will occur
 - b) It is a non-terminal whose production will be used for reduction in the next step
 - c) It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur.
 - d) It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found

Answer: d
 Explanation: Handles are the part of sentential form, & they are identified as the right side of any given production which will be used for reduction in the next step.

6. Some code optimizations are carried out on the intermediate code because _____

- a) They enhance the portability of the complier to other target processors
- b) Program analysis is name accurate on intermediate code than on machine code
- c) The information from data flow analysis cannot otherwise be used for optimization
- d) The information from the front end cannot otherwise be used for optimization

Answer: b
 Explanation: Code optimizations are carried out on the intermediate code because program analysis is more accurate on intermediate code than on machine code. Answer: a

Explanation: I. Statement is false since global variables are required for recursions with static storage. This is due to unavailability of stack in static storage. II. This is true III. In dynamic allocation heap structure is used, so it is false. IV. False since recursion can be implemented. V. Statement is completely true. So only II & V are true.

- 8. An LALR(1) parser for a grammar can have shift-reduce (S-R) conflicts if and only if _____**
- a) The SLR(1) parser for G has S-R conflicts**
 - b) The LR(1) parser for G has S-R conflicts**
 - c) The LR(0) parser for G has S-R conflicts**
 - d) The LALR(1) parser for G has reduce-reduce conflicts**

Answer: b
Explanation: LALR parser is reduced form of CLR or LR(1) parser, LALR parser uses the LR(1) items of CLR parser & of any shift reduce conflicts are there then it is due to LR(1) parser.
Answer: b
Explanation: I. Statement is true since there are some parsers which take $O(n \log 2n)$ times for parsing. II. Completely false, since there is no use of stack which is required for recursion. III. False IV. True since both types of optimizations are applied.

- 10. What data structure in a compiler is used for managing information about variables and their attributes?**
- a) Abstract syntax tree**
 - b) Symbol table**
 - c) Semantic stack**
 - d) Parse table**

Answer: b
Explanation: Symbol table is used for storing the information about variables and their attributes by compiler.

- 1. Which languages necessarily need heap allocation in the runtime environment?**
- a) Those that support recursion**
 - b) Those that use dynamic scoping**
 - c) Allow dynamic data structure**
 - d) Those that use global variables**

Answer: c
Explanation: E.g.: Heap.

- 2. Given the language $L = \{ab, aa, baa\}$, which of the following strings are in L^* ?**

- 1) abaabaaabaa
- 2) aaaabaaaa
- 3) baaaaabaaaab
- 4) baaaaabaa

- a) 1,2 and 3**
- b) 2,3 and 4**
- c) 1,2 and 4**
- d) 1,3 and 4**

Answer: c
*Explanation: Generates strings like aaaabaaaa
baaaaabaa*

- 3. The lexical analyzer takes _____ as input and produces a stream of _____ as output.**
- a) Source program, tokens**
 - b) Token, source program**
 - c) Either A and B**
 - d) None of the mentioned**

Answer: a
Explanation: Lexical analyser takes source program as input and token as output.

4. The action of parsing the source program into proper syntactic classes is called _____

- a) Syntax analysis**
- b) Lexical analysis**
- c) Interpretation analysis**
- d) General syntax analysis**

Answer: b

Explanation: Checks for correct syntax.

5. What is the task of the lexical analysis?

- a) None of the mentioned**
- b) To build a literal and identifier table**
- c) To build a uniform symbol table**
- d) To build a uniform symbol table, literal and identifier table**

Answer: d

Explanation: It is the task performed.

1. In a two pass assembler, adding literals to literal table and address resolution of local symbols are done using?

- a) First pass and second respectively**
- b) Both second pass**
- c) Second pass and first respectively**
- d) Both first pass**

Answer: d

Explanation: A two pass assembler does two passes over the source file (the second pass can be over a file generated in the first pass).

2. In Two pass assembler the object code generation is done during the _____

- a) Second pass**
- b) First pass**
- c) Zeroth pass**
- d) Not done by assembler**

Answer: a

Explanation: On the second pass, the assembler:

- *source statements into machine code*
- *error messages, if error has occurred.*

3. Pick the machine independent phase of the compiler:

- a) Syntax analysis**
- b) Code generation**
- c) Lexical analysis**
- d) All of the mentioned**

Answer: d

Explanation: Machine independent phases are Lexical analysis, Syntax analysis, Semantic analysis, Intermediate code generation and sometime code optimization.

4. A system program that combines the separately compiled modules of a program into a form suitable for execution is?

- a) Assembler**
- b) Linking loader**
- c) Cross compiler**
- d) Load and Go**

Answer: b

Explanation: Combines the modules which have been compiled separately.

5. Which of the following type of software should be used if you need to create, edit and print documents?

- a) Word processing
- b) Spreadsheet
- c) Desktop publishing
- d) UNIX

Answer: an

Explanation: Application software such as word processors.

6. Output file of the Lex is _____ is the input file is Sam.

- a) sam
- b) sam.yy.c
- c) sam.lex
- d) sam.obj

Answer: b

Explanation: This Produce the file “sam.yy.c”, which we can then compile with g++.

7. Type checking is normally done during _____

- a) Lexical analysis
- b) Syntax analysis
- c) Syntax directed translation
- d) Code generation

Answer: c

Explanation: It enables the compiler to do type checking.

8. Yacc is available as a command on the _____

- a) MINIX
- b) UNIX
- c) DOS
- d) None of the mentioned

Answer: b

Explanation: Unix provides with a YACC command.

9. Loading process can be divided into two programs. The first is binder the other is _____

- a) Linkage editor
- b) Module Loader
- c) Relocate
- d) None of the mentioned

Answer: b

Explanation: A module loader is an answer.

10. In Lex, a class is complemented by first placing _____

- a) ^
- b) OR
- c) –
- d) NOT

Answer: a

Explanation: ^ =complement.

1. Which of the following is not a feature of the compiler?

- a) Converts it into machine code
- b) None of the mentioned
- c) Slow for debugging

d) Execution time is more

Answer: d

Explanation: No specified execution time.

2. A programming language is to be designed to run on a machine that does not have a big memory.

- a) Prefer a 2 pass compiler**
- b) Prefer an interpreter**
- c) Not support recursion**
- d) All of the mentioned**

Answer: d

Explanation: If a language designed to run on a machine then recursion should not be followed it should prefer 2 pass assembler and prefer an interpreter to a compiler.

3. What is linker?

- a) Is same as the loader**
- b) Create a load module**
- c) Is same as the loader & Create a load module**
- d) None of the mentioned**

Answer: b

Explanation: It creates a module to be loaded.

4. Predictive parsers can be _____

- a) Recursive**
- b) Constructive**
- c) Non recursive**
- d) Both Recursive and Constructive**

Answer: a

Explanation: The nature of predictive parsers can be Recursive.

5. Producer consumer problem can be solved using _____

- a) Semaphores**
- b) Event counters**
- c) Monitors**
- d) All of the mentioned**

Answer: d

Explanation: The famous producer consumer problem can be solved by a semaphores event counters as well as monitors.

6. Bottom up parsing involves _____

- a) Shift reduce**
- b) Handle pruning**
- c) Shift reduce & Handle pruning**
- d) None of the mentioned**

Answer: c

Explanation: The bottom up parsing involves Shift reduce operation and Handle pruning.

7. What is an example of intermediate language?

- a) SNOBOL**
- b) PASCAL**
- c) COBOL**
- d) UNCOL**

Answer: d

Explanation: Others are HLL.

- 8. In a two pass assembler the object code generation is done during the _____**
- a) Second pass**
 - b) First pass**
 - c) Zeroth pass**
 - d) Not done by assembler**

Answer: a

Explanation: I second pass object code is generated.

- 1. In which addressing mode the operand is given explicitly in the instruction?**
- a) Absolute mode**
 - b) Immediate mode**
 - c) Indirect mode**
 - d) Index mode**

Answer: b

Explanation: In Immediate addressing mode operand is directly found from the instruction.

- 2. Which of the following is not true?**
- a) UGL doesn't support high level seven interaction**
 - b) Many DBMS support 4GLs**
 - c) All of the mentioned**
 - d) None of the mentioned**

Answer: a

Explanation: It supports HLL.

- 3. Non modifiable procedures are called _____**
- a) Concurrent procedures**
 - b) Serially usable procedures**
 - c) Re-entrant procedures**
 - d) Top down procedures**

Answer: b

Explanation: These can't be modified.

- 4. Which of the following is not a type of assembler?**
- a) One pass**
 - b) Two pass**
 - c) Three pass**
 - d) Load and go**

Answer: c

Explanation: No 3 pass assembler created as yet.

- 5. Daisy chain is a device for _____**
- a) None of the mentioned**
 - b) Connecting devices to a controller**
 - c) Connecting controller to devices**
 - d) All of the mentioned**

Answer: b

Explanation: Combines multiple devices in sequence or a ring.

- 6. What is the input of Lex?**
- a) Set to regular expression**

- b) Statement
- c) Numeric data
- d) ASCII data

Answer: a

Explanation: Input is a string.

7. Yacc semantic action is a sequence of _____

- a) Tokens
- b) Expression
- c) Statement
- d) Rules

Answer: c

Explanation: Yacc is yet another compiler which is a set of statements.

8. Which of the following software tool is parser generator?

- a) Lex
- b) Yacc
- c) Lex & Yacc
- d) None of the mentioned

Answer: b

Explanation: YACC is a LALR parser.

9. A Lex compiler generates _____

- a) Lex object code
- b) Transition tables
- c) Tokens
- d) None of the mentioned

Answer: b

Explanation: The output of Lex is a string of tokens.

10. A Compiler has _____ phases.

- a) 7
- b) 6
- c) 8
- d) None of the mentioned

Answer: c

Explanation: It has 8 phases.

- a) 1
- b) 2
- c) 3
- d) unknown/undefined

Answer: a

Explanation: b will be one because when value of a is 1 it is stored.

2. What value does the variable a have after ALL of the code above executes?

```
int a;  
int b;  
a=1;  
b=a++;
```

- a) 1

- b) 2
- c) 3
- d) unknown/undefined

Answer: b

Explanation: a=2 cause it has been incremented.

3. What value does the variable z have after ALL of the code above executes?

```
int a;  
int b;  
a=1;  
b=a++;
```

- a) 9
- b) 12
- c) 16
- d) 20

Answer: c

Explanation: z=4 4*

Hence the answer will be 16.

4. What value does the variable x have after ALL of the code above executes?

```
int x;  
int y;  
int z;  
x=3;  
y=4;  
z = ++x * y++;
```

- a) 2
- b) 3
- c) 4
- d) unknown/undefined

Answer: c

Explanation: Finally the value of x is 4.

5. What value does the variable y have after ALL of the code above executes?

```
int x;  
int y;  
int z;  
x=3;  
y=4;  
z = ++x * y++;
```

- a) 4
- b) 5
- c) 6
- d) unknown/undefined

Answer: b

Explanation: The value of y is increased by 1 and becomes 5.

6. What will be the output of the following program?

```
int x;  
int y;  
int z;  
x=3;
```

```
y=4;  
z = ++x * y++;
```

- a) 11
- b) 10
- c) -9
- d) None of the mentioned

Answer: d

Explanation: The expression ++(-i) is not valid but -(++i) is valid.

7. What will be output of the following C code?

```
#include<stdio.h>  
int main()  
{  
    int i = 10;  
    printf("%d", ++(-i));  
    return 0;  
}
```

- a) 4,3,3
- b) 3,3,3
- c) 2,3,3
- d) 4,4,3

Answer: c

Explanation: y = 3 and z= 3 but x has decremented and become 2.

8. What will be output of the following C code?

```
#include <stdio.h>  
int main()  
{  
    int x=4, y, z;  
    y = --x;  
    z = x--;  
    printf("%d, %d, %d\n", x, y, z);  
    return 0;  
}
```

- a) a = 6, b = 15
- b) a = 1, b = 3
- c) a = 1, b = 15
- d) a = 2, b = 4

Answer: a

Explanation: B=1+2+3+4+5

B=15

But finally a=6.

9. What will be output of the following C code?

```
#include <stdio.h>  
main()  
{  
    int a=1, b=3;  
    b= a++ + a++ + a++ + a++ + a++;  
    printf("a=%d \n b=%d",a,b);  
}
```

- a) 9,9

- b) 10,10
- c) 9,10
- d) 10,9

Answer: b

Explanation: A=9

B=9

B=10

10. What will be output of the following C code?

```
#include <stdio.h>
main()
{
    int a=9, b=9;
    a=b++;
    b=a++;
    b=++b;
    printf("%d %d", a, b);
}
```

- a) 24,12
- b) 23,12
- c) 23,10
- d) 24,10

Answer: b

Explanation: A = 11+12

So a=23

B=12.

1. In C programming language, which of the following type of operators have the highest precedence?

- a) Relational Operators
- b) Equality Operators
- c) Logical Operators
- d) Arithmetic Operators

Answer: d

Explanation: No other operator has higher precedence than arithmetic operator.

2. Which of the following operators has the highest precedence?

- a) Unary +
- b) *
- c) >=
- d) ==

Answer: a

Explanation: Unary operators have max precedence in over all other arithmetic operators.

3. If i=1 j=2,k=3, then what is the value of the following expression?

```
!((j + k) > (i + 5))
```

- a) 6
- b) 5
- c) 1
- d) 0

Answer: c

Explanaton: !((2+3)>(1+5))

!(5>6)

Since the condition is false hence !(0)

And complement of 0 is 1.

4. The expression $5 - 2 - 3 * 5 - 2$ will evaluate to 18, if $-$ is left associative and _____

- a) $*$ has precedence over $-$**
- b) $*$ has precedence over $-$**
- c) $-$ has precedence over $*$**
- d) $-$ has precedence over $-$**

Answer: c

Explanation: if $-$ has precedence over $*$ and if it associates from the right.

5. Coercion _____

- a) Takes place over an assignment operator**
- b) Operator has operands of various types**
- c) None of the mentioned**
- d) Takes place over an assignment operator & Operator has operands of various types**

Answer: d

Explanation: Conversion between compatible types.

6. Choose the correct statement.

- a) Expressions evaluated at compile time**
- b) String constants concatenated at compile time**
- c) None of the mentioned**
- d) Both of the mentioned**

Answer: d

Explanation: The statements are true.

7. Which of the following operators takes only integer operands?

- a) $+$**
- b) $*$**
- c) $/$**
- d) $\%$**

Answer: d

Explanation: Two integers are taken to be input.

8. Pick the operators that associate from the left.

- a) $+$**
- b) $,$**
- c) $<$**
- d) All of the mentioned**

Answer: d

Explanation: They are left associative.

9. Pick the operators that associate from the right.

- a) $?:$**
- b) $+=$**
- c) $=$**
- d) All of the mentioned**

Answer: d

Explanation: They are right associative.

10. Pick the operators that associate from left to right.

- a) &&
- b) ?:
- c) ,
- d) All of the mentioned

Answer: d

Explanation: They left to right associative.

1. Which of the following correctly declares an array?

- a) `Int array[10]`
- b) `int array`
- c) `array{10}`
- d) `array array[10]`

Answer: a

Explanation: Correct declaration.

2. What is the index number of the last element of an array with 29 elements?

- a) 29
- b) 28
- c) 0
- d) Programmer-Defined

Answer: b

Explanation: The indexing in an array starts with zero hence we can say that the element.

3. Which of the following is a two-dimensional array?

- a) `array array[20][20]`
- b) `int array[20][20]`
- c) `int array[20, 20]`
- d) `char array[20]`

Answer: b

Explanation: Double dimensional arrays are declared in this format.

4. Which of the following correctly accesses the seventh element stored in tan?

- a) `tan[6]`
- b) `tan[7]`
- c) `tan(7)`
- d) `tan`

Answer: a

Explanation: The index no 6.

5. Which of the following gives the memory address of the first element in array tan?

- a) `tan[0]`
- b) `tan`
- c) `&tan`
- d) `tan [1]`

Answer: b

Explanation: The base address of the array is given by its name.

6. What will happen if in a C program you assign a value to an array element whose subscript exceeds the size of array?

- a) The compiler would report an error
- b) May stop working abruptly if data gets overwritten

- c) None of the mentioned
d) The element will be set to 0

Answer: b

Explanation: It often happens that the program crashes.

7. What does the following declaration mean?

```
int (*a)[10] a
```

- a) Pointer to an array
b) None of the mentioned
c) Array of 10 integers
d) Pointer to an array & Array of 10 integers

Answer: a

Explanation: Points to array.

8. What is the meaning of the following declaration?

```
Int arr[20]
```

- a) Integer Array of size 20
b) None of the mentioned
c) Array of size 20
d) Array of size 20 that can have higher integer address

Answer: a

Explanation: Declaration of an array.

9. What will be the size of below array elements?

```
int a[20]
```

- a) 21
b) 22
c) 20
d) 19

Answer: c

Explanation: The number in square brackets denotes size of an array.

10. What is meaning of the following?

```
Int *ptr[20]
```

- a) Integer array of size 20 pointing to an integer Pointer
b) None of the mentioned
c) Array of integer pointer of size 20
d) All of the mentioned

Answer: c

Explanation: Array of pointers to integers.

1. In which way can SAM be stored in an array?

- a) char name[]="SAM"
b) char name[6]={'S','A','M'}
c) char name[6]={"S","A","M"}
d) char name [6]={'S','A','M','\0'}

Answer: d

Explanation: The array consists of the word SAM and also a NULL character.

2. Int RollNUm[30][4] is a?

- a) 1-d Array**
- b) 3-D Array**
- c) 4- D Array**
- d) 2-D Array**

Answer: d

Explanation: 2 [] brackets means a double dimensional array.

3. What will be output of the following code?

```
#include<stdio.h>
void main()
{
    char arr[7]="Network";
    printf("%s",arr);
}
```

- a) Network**
- b) N**
- c) network**
- d) Garbage Value**

Answer: d

Explanation: Garbage value is printed.

4. What will be output of the following code?

```
#include<stdio.h>
void main()
{
    char arr[11]="The African Queen";
    printf("%s",arr);
}
```

- a) The African Queen**
- b) The**
- c) Queen**
- d) Null**

Answer: d

Explanation: It gives a Null Value.

5. What will be output if you will execute following C code?

```
#include<stdio.h>
void main()
{
    char arr[20]="MysticRiver";
    printf("%d",sizeof(arr));
}
```

- a) 20**
- b) 11**
- c) 30**
- d) None of the mentioned**

Answer: a

Explanation: The size of array is 20.

6. What will be output if you will execute following C code?

```
#include<stdio.h>
void main(){
    int const SIZE=5;
    int expr;
    double value[SIZE]={2.0,4.0,6.0,8.0,10.0};
    expr=1|2|3|4;
    printf("%f",value[expr]);
}
```

- a) 2.00000
- b) 4.00000
- c) 6.00000
- d) Compilation error

Answer: d
Explanation: No fixed value in [] brackets.

7. Comment on the following statement:

```
int (*a)[7];
```

- a) An array “a” of pointers
- b) A pointer “a” to an array
- c) A ragged array
- d) None of the mentioned

Answer: b
Explanation: It is a pointer to array a.

8. Comment on the 2 arrays regarding P and Q:

```
int *a1[8];
int *(a3[8]);
P. Array of pointers
Q. Pointer to an array
```

- a) a1 is P, a2 is Q
- b) a1 is P, a2 is P
- c) a1 is Q, a2 is P
- d) a1 is Q, a2 is Q

Answer: b
Explanation: p =a1
And a3 is alo p.

8. Which of the following is not possible statically in C?

- a) Jagged Array
- b) Rectangular Array
- c) Cuboidal Array
- d) Multidimensional Array

Answer: a
Explanation: Jagged array is not possible.

9. What is the max no. of dimensions an array may have?

- a) 2
- b) 8
- c) 20
- d) No limit

Answer: d

Explanation: Dimension of an array can be any number.

10. Array passed as an argument to a function is interpreted as _____

- a) Address of the array**
- b) None of the mentioned**
- c) Base address**
- d) No. of elements of the array**

Answer: c

Explanation: The base address is passed when the array is passed as an argument in a function.

1. Select the output for following set of code.

```
static void Main(string[] args)
{
    int movie = 1;
    switch (movie <<< 2 + movie)
    {
        default:
            Console.WriteLine("A");
            break;
        case 4:
            Console.WriteLine("B");
            break;
        case 5:
            Console.WriteLine("C");
            break;
        case 8:
            Console.WriteLine("D");
            break;
    }
    Console.ReadLine();
}
```

- a) A**
- b) B**
- c) C**
- d) D**

Answer: c

Explanation: 'default' case can be put in anywhere.

Output: D.

2. Select the output for following set of code.

```
static void Main(string[] args)
{
    int i = 2, j = 4;
    switch (i + j * 2)
    {
        case 1 :
        case 2 :
            Console.WriteLine("1 and 2");
            break;
        case 3 to 10:
            Console.WriteLine("3 to 10");
            break;
    }
    Console.ReadLine();
}
```

- a) 3 to 10 will be printed**
- b) 1 and 2 will be printed**

c) error

d) The code gives output as 3 to 10

Answer: c

Explanation:

Output:

Here $i = 2, j = 4$.

3. Select the output for following set of code.

```
static void Main(string[] args)
{
    int i = 2, k = 3;
    switch (i - k)
    {
        case -1:
            ++i;
            ++k;
            break;
        case 2:
            --i;
            ++k;
            break;
        default:
            i += 3;
            k += i;
            break;
    }
    Console.WriteLine(i + "\n" + k);
    Console.ReadLine();
}
```

a) 2 3 3

b) 3 2 3

c) 3 4 4

d) 5 10 10

Answer: c

Output: 3

4

4

Explanation: $i - k = -1$. So, case -1 will be executed only.

4. Select output for following set of code.

```
static void Main(string[] args)
{
    int const p = 0;
    switch (3 * 5 / 6)
    {
        case p:
            Console.WriteLine("A");
            break;
        case p * 1:
            Console.WriteLine("B");
            break;
        case p - 2:
            Console.WriteLine("C");
            break;
        default:
            Console.WriteLine("D");
    }
}
```

a) A

- b) B
- c) C
- d) Compile time error

Answer: d

Explanation: No constant variable.

5. Select output for following set of code.

```
static void Main(string[] args)
{
    int i = 2, j = 3, k = 4;
    switch (i + j - k)
    {
        case 0: case 2: case 4:
            ++i;
            k += j;
            break;
        case 1: case 3: case 5 :
            --i;
            k -= j;
            break;
        default:
            i += j;
            break;
    }
    Console.WriteLine(i + "\n" + j + "\n" + k);
    Console.ReadLine();
}
```

- a) 1 3 1
- b) 2 3 4
- c) 5 3 4
- d) Compile time error

Answer: a

Explanation: Solving expression gives 1

Output: 1

3

1

6. Select the output for following set of code.

```
static void Main(string[] args)
{
    int i = 9 , j = 7;
    switch (i - j + 3)
    {
        case 9: 7:
            j += 6;
            break;
        case 5:
            i -= 4;
            break;
    }
    Console.WriteLine(i + "\n" + j);
    Console.ReadLine();
}
```

- a) 5 7
- b) 9 13
- c) Compile time error
- d) 9 7

Answer: c

Explanation: Invalid expression.

7. Select the output for code :

```
static void Main(string[] args)
{
    switch (5)
    {
        case 5.0f:
            Console.WriteLine("harsh");
            break;
        case 5:
            Console.WriteLine("amish");
            break;
        case 5.0L:
            Console.WriteLine("ANKIT");
            break;
        default:
            Console.WriteLine("ashish");
    }
    Console.ReadLine();
}
```

- a) amish
- b) ANKIT
- c) harsh
- d) Compile time error

Answer: d

Explanation: Only integers are allowed .

5.0f = (int)5.0f.

5.0L = (int)5.0L.

8. Select output for code.

```
static void Main(string[] args)
{
    int i;
    int j = 1;
    int []ar = {21, 22, 13, 4};
    switch (ar[j])
    {
        case 1:
            i++;
            break;
        case 2:
            i += 2;
            j = 3;
            continue;
        case 3:
            i %= 2;
            j = 4;
            continue;
        default:
            --i;
    }
    Console.WriteLine(i);
    Console.ReadLine();
}
```

- a) 23
- b) 15
- c) Compile time error
- d) 12

Answer: c

Explanation: Continue cannot be used.

9. Select the output for following set of Code.

```
static void Main(string[] args)
{
    char ch = Convert.ToChar('a' | 'b' | 'c');
    switch (ch)
    {
        case 'A':
        case 'a':
            Console.WriteLine("case A|case a");
            break;
        case 'B':
        case 'b':
            Console.WriteLine("case B|case b");
            break;
        case 'C':
        case 'c':
        case 'D':
        case 'd':
            Console.WriteLine("case D|case d");
            break;
    }
    Console.ReadLine();
}
```

a) Compile time error

b) case A | case a

c) case B | case b

d) case D | case d

Answer: d

Explanation: case D|case d

10. Select the output for following set of Code.

```
static void Main(string[] args)
{
    char ch = 'p';
    switch (ch)
    {
        case 'p':
            Console.WriteLine("coco" + "\t" + Convert.ToInt32(ch));
            break;
        default:
            Console.WriteLine("default");
            break;
    }
    Console.WriteLine("main");
}
```

a) coco main

b) coco 112

c) coco 112 main

d) compile time error

Answer: c

Explanation: ASCII value of p is 112.

Output: coco 112 main.

1. How many times will the following loop be executed?

```
ch = 'b';
```



```
while(ch >= 'a' && ch <= 'z')
```

- a) 0
- b) 25
- c) 26
- d) 1

Answer: b

Explanation: Because there are 26 alphabets and so it runs for 26-1 times.

2. Consider the following program.

```
switch(input)
{
    case '1':
        printf("One");
    case '2':
        printf("Two");
    case '3':
        printf("Three");
    default:
        printf("Default");
        break;
}
```

What will be printed when input is 2?

- a) Two Three default
- b) Two
- c) Two default
- d) Two Two default

Answer: a

Explanation: Because in between cases there is no break statement.

Answer: a
Explanation: Switch is more convenient to use as it avoids confusion in case of more nested structure.

4. Consider the following C program. What is the Value of a?

```
#include<stdio.h>
int main()
{
    int a=7, b=5;
    switch(a = a % b)
    {
        case 1:
            a = a - b;
        case 2:
            a = a + b;
        case 3:
            a = a * b;
        case 4:
            a = a / b;
        default:
            a = a;
    }
    return 0;
}
```

- a) 7
- b) 5
- c) 9
- d) None of the mentioned

Answer: a

Explanation: Initially a=a%b gives 2. Then in case 2 a=a+b gives 2+5=7. Hence a=7.

Answer: b

Explanation: Infinite loop.

6. What will be the output of the following Code?

```
void main()  
{  
    static a,b;  
    while(a > b++)  
}
```

- a) a=0 b=0**
- b) a=0 b=0**
- c) a=1 b=1**
- d) none of the mentioned**

Answer: a

Explanation: Static variables have an initial value 0.

7. Number of times the loop will run.

```
for(digit = 0;digit &lt; 9; digit++)  
{  
    digit = digit *2;  
    digit--;  
}
```

- a) Infinite**
- b) 9**
- c) 0**
- d) 4**

Answer: a

Explanation: Infinite times it runs.

Answer: b

Explanation: If can also have compound statements.

Answer: a

Explanation: Can be converted to for loop.

Answer: b

Explanation: Default cases may or may not be written.

1. The load instruction is mostly used to designate a transfer from memory to a processor register known as _____

- a) Accumulator**
- b) Instruction Register**
- c) Program counter**
- d) Memory address Register**

Answer: a

Explanation: Accumulator is the process register.

2. A group of bits that tell the computer to perform a specific operation is known as _____

- a) Instruction code**
- b) Micro-operation**
- c) Accumulator**
- d) Register**

Answer: a

Explanation: Instruction code is the set of specific tasks to be performed.

3. The time interval between adjacent bits is called the _____

- a) Word-time**
- b) Bit-time**
- c) Turnaround time**
- d) Slice time**

Answer: b

Explanation: Bit time is the interval time here.

4. A k-bit field can specify any one of _____

- a) 3k registers**
- b) 2k registers**
- c) K2 registers**
- d) K3 registers**

Answer: b

Explanation: 2k is the answer.

5. MIMD stands for _____

- a) Multiple instruction multiple data**
- b) Multiple instruction memory data**
- c) Memory instruction multiple data**
- d) Multiple information memory data**

Answer: a

Explanation: MIMD stands for multiple instructions multiple data.

6. Logic gates with a set of input and outputs are arrangement of _____

- a) Computational circuit**
- b) Logic circuit**
- c) Design circuits**
- d) Register**

Answer: a

Explanation: The answer is computational circuit.

7. The average time required to reach a storage location in memory and obtain its contents is called _____

- a) Latency time**
- b) Access time**
- c) Turnaround time**
- d) Response time**

Answer: b

Explanation: Access time is the time from the start of one storage device access to the time when the next access can be started.

8. The BSA instruction is _____

- a) Branch and store accumulator**
- b) Branch and save return address**
- c) Branch and shift address**
- d) Branch and show accumulator**

Answer: b

Explanation: BSA is branch and save return address.

9. A floating point number that has an O in the MSB of mantissa is said to have _____

- a) Overflow**
- b) Underflow**
- c) Important number**
- d) Undefined**

Answer: b

Explanation: It is an underflow condition.

10. Translation from symbolic program into Binary is done in _____

- a) Two passes
- b) Directly
- c) Three passes
- d) Four passes

Answer: a

Explanation: Two passes are required.

1. The instruction 'ORG O' is a _____

- a) Machine Instruction
- b) Pseudo instruction
- c) High level instruction
- d) Memory instruction

Answer: b

Explanation: It is a pseudo instruction.

2. 'Aging registers' are _____

- a) Counters which indicate how long ago their associated pages have been Referenced
- b) Registers which keep track of when the program was last accessed
- c) Counters to keep track of last accessed instruction
- d) Counters to keep track of the latest data structures referred

Answer: a

Explanation: This is known as aging.

3. Memory unit accessed by content is called _____

- a) Read only memory
- b) Programmable Memory
- c) Virtual Memory
- d) Associative Memory

Answer: d

Explanation: Associative Emory is accessed by content.

4. _____ register keeps tracks of the instructions stored in program stored in memory.

- a) AR (Address Register)
- b) XR (Index Register)
- c) PC (Program Counter)
- d) AC (Accumulator)

Answer: c

Explanation: Program counter keeps a track.

5. N bits in operation code imply that there are _____ possible distinct Operators.

- a) 2n
- b) 2ⁿ
- c) n/2
- d) n²

Answer: b

Explanation: 2ⁿ possible combinations.

6. A three input NOR gate gives logic high output only when _____

- a) One input is high
- b) One input is low
- c) Two input are low

d) All input are high

Answer: d

Explanation: All inputs are high.

7. The circuit converting binary data in to decimal is _____

- a) Encoder**
- b) Multiplexer**
- c) Decoder**
- d) Code converter**

Answer: d

Explanation: It is the code converter.

8. The multiplicand register & multiplier register of a hardware circuit implementing booth's algorithm have (11101) & (1100). The result shall be _____

- a) (812)10**
- b) (-12)10**
- c) (12)10**
- d) (-812)10**

Answer: a

Explanation: On multiplying we get 81210.

9. PSW is saved in stack when there is a _____

- a) Interrupt recognized**
- b) Execution of RST instruction**
- c) Execution of CALL instruction**
- d) All of the mentioned**

Answer: a

Explanation: It can be interrupted.

10. In computers, subtraction is carried out generally by _____

- a) 1's complement method**
- b) 2's complement method**
- c) signed magnitude method**
- d) BCD subtraction method**

Answer: b

Explanation: Subtraction done by 2's complement.

1. The main memory in a Personal Computer (PC) is made of _____

- a) Cache memory**
- b) Static RAM**
- c) Dynamic Ram**
- d) Cache memory & RAM**

Answer: d

Explanation: It is made up of cache memory and RAM.

2. Cache memory works on the principle of _____

- a) Locality of data .**
- b) Locality of memory**
- c) Locality of reference**
- d) Locality of reference & memory**

Answer: c

Explanation: Cache works on Locality of reference.

- 3. An n-bit microprocessor has _____**
- a) n-bit program counter**
 - b) n-bit address register**
 - c) n-bit ALU**
 - d) n-bit instruction register**

Answer: d

Explanation: N bit microprocessor has bit instruction register.

- 4. When CPU is executing a Program that is part of the Operating System, it is said to be in _____**
- a) Interrupt mode**
 - b) System mode**
 - c) Half mode**
 - d) Simplex mode**

Answer: b

Explanation: It is in System Mode.

- 5. Logic X-OR operation of (4ACO) H & (B53F) H results _____**
- a) AACB**
 - b) 0000**
 - c) FFFF**
 - d) ABCD**

Answer: c

Explanation: Xoring the two operands we get FFFF.

- 6. If the main memory is of 8K bytes and the cache memory is of 2K words. It uses associative mapping. Then each word of cache memory shall be _____**
- a) 11 bits**
 - b) 21 bits**
 - c) 16 bits**
 - d) 20 bits**

Answer: c

Explanation: Each should be $8 \times 2 = 16$ bits.

- 7. A Stack-organised Computer uses instruction of _____**
- a) Indirect addressing**
 - b) Two-addressing**
 - c) Zero addressing**
 - d) Index addressing**

Answer: c

Explanation: Zero addressing modes are used in stack organised computer.

- 8. In a program using subroutine call instruction, it is necessary _____**
- a) Initialize program counter**
 - b) Clear the accumulator**
 - c) Reset the microprocessor**
 - d) Clear the instruction register**

Answer: d

Explanation: It is mandatory to clear the instruction register.

- 9. Virtual memory consists of _____**

- a) Static RAM
- b) Dynamic RAM
- c) Magnetic memory
- d) None of the mentioned

Answer: a

Explanation: It does consist of static RAM.

10. In signed-magnitude binary division, if the dividend is $(11100)_2$ and divisor is $(10011)_2$ then the result is

-
- a) $(00100)_2$
 - b) $(10100)_2$
 - c) $(11001)_2$
 - d) $(01100)_2$

Answer: b

Explanation: After division we get 20 which is 10100 .

1. Generally Dynamic RAM is used as main memory in a computer system as it _____

- a) Consumes less power
- b) Has higher speed
- c) Has lower cell density
- d) Needs refreshing circuitry

Answer: b

Explanation: Because of higher speed it is Dynamic.

2. Write Through technique is used in which memory for updating the data _____

- a) Virtual memory
- b) Main memory
- c) Auxiliary memory
- d) Cache memory

Answer: d

Explanation: The answer is cache memory.

3. Cache memory acts between _____

- a) CPU and RAM
- b) RAM and ROM
- c) CPU and Hard Disk
- d) None of the mentioned

Answer: a

Explanation: It acts between CPU and RAM.

4. The circuit used to store one bit of data is known as _____

- a) Encoder
- b) OR gate
- c) Flip Flop
- d) Decoder

Answer: c

Explanation: Flip flop is 1 bit circuit.

5. Von Neumann architecture is _____

- a) SISD
- b) SIMD
- c) MIMD

d) MISD

Answer: a

Explanation: It is single instruction single data.

6. In a vectored interrupt _____

- a) The branch address is assigned to a fixed location in memory**
- b) The interrupting source supplies the branch information to the processor through an interrupt vector**
- c) The branch address is obtained from a register in the processor**
- d) None of the mentioned**

Answer: b

Explanation: It branches to process the interrupt.

7. In a memory-mapped I/O system, which of the following will not be there?

- a) LDA**
- b) IN**
- c) ADD**
- d) OUT**

Answer: a

Explanation: There is no LDA.

8. If memory access takes 20 ns with cache and 110 ns without it, then the ratio (cache uses a 10 ns memory) is

-
- a) 93%**
 - b) 90%**
 - c) 88%**
 - d) 87%**

Answer: b

Explanation: The answer is 90%.

9. The addressing mode used in an instruction of the form ADD X Y is _____

- a) Absolute**
- b) Indirect**
- c) Index**
- d) None of the mentioned**

Answer: c

Explanation: This addressing mode is indexed.

10. _____ register keeps track of the instructions stored in program stored in memory.

- a) AR (Address Register)**
- b) XR (Index Register)**
- c) PC (Program Counter)**
- d) AC (Accumulator)**

Answer: c

Explanation: Program Counter keeps track of the next instruction.

1. The idea of cache memory is based _____

- a) On the property of locality of reference**
- b) On the heuristic 90-10 rule**
- c) On the fact that references generally tend to cluster**
- d) All of the mentioned**

Answer: a

Explanation: Cache memory is based on the locality of references.

2. Which of the following is not a weighted code?

- a) Decimal Number system**
- b) Excess 3-cod**
- c) Binary number System**
- d) None of the mentioned**

Answer: b

Explanation: Excess 3 is not a weighted code.

3. The average time required to reach a storage location in memory and obtain its contents is called the _____

- a) Seek time**
- b) Turnaround time**
- c) Access time**
- d) Transfer time**

Answer: c

Explanation: Times used to access the contents.

4. (2FAOC) 16 is equivalent to _____

- a) (195 084)₁₀**
- b) (001011111010 0000 1100)₂**
- c) (195 084)₁₀ & (001011111010 0000 1100)₂**
- d) None of the mentioned**

Answer: b

Explanation: It is equivalent to (001011111010 0000 1100)₂.

5. The circuit used to store one bit of data is known as _____

- a) Register**
- b) Encoder**
- c) Decoder**
- d) Flip Flop**

Answer: d

Explanation: 1 bit circuit is known as Flip Flop.

6. Computers use addressing mode techniques for _____

- a) Giving programming versatility to the user by providing facilities as pointers to memory counters for loop control**
- b) To reduce number of bits in the field of instruction**
- c) Specifying rules for modifying or interpreting address field of the instruction**
- d) All of the mentioned**

Answer: d

Explanation: All of these are addressing mode techniques.

7. What characteristic of RAM memory makes it not suitable for permanent storage?

- a) Too slow**
- b) Unreliable**
- c) It is volatile**
- d) Too bulky**

Answer: c

Explanation: Ram is volatile.

8. The amount of time required to read a block of data from a disk into memory is composed of seek time, rotational latency, and transfer time. Rotational latency refers to _____

- a) The time it takes for the platter to make a full rotation
- b) The time it takes for the read-write head to move into position over the appropriate track
- c) The time it takes for the platter to rotate the correct sector under the head
- d) None of the mentioned

Answer: a

Explanation: Rotational latency is the time taken to make full rotation.

1. Suppose that a bus has 16 data lines and requires 4 cycles of 250 nsecs each to transfer data. The bandwidth of this bus would be 2 Megabytes/sec. If the cycle time of the bus was reduced to 125 nsecs and the number of cycles required for transfer stayed the same what would the bandwidth of the bus?

- a) 1 Megabyte/sec
- b) 4 Megabytes/sec
- c) 8 Megabytes/sec
- d) 2 Megabytes/sec

Answer: d

Explanation: The bandwidth is 2 mb/s.

2. Floating point representation is used to store _____

- a) Boolean values
- b) Whole numbers
- c) Real integers
- d) Integers

Answer: c

Explanation: They are real Integers.

3. SIMD represents an organization that _____

- a) Refers to a computer system capable of processing several programs at the same time
- b) Represents organization of single computer containing a control unit, processor unit and a memory unit
- c) Includes many processing units under the supervision of a common control unit
- d) None of the mentioned

Answer: c

Explanation: SIMD includes processing units under the supervision of a common control.

4. In Reverse Polish notation, expression $A*B+C*D$ is written as _____

- a) $AB*CD*+$
- b) $A*BCD*+$
- c) $AB*CD+*$
- d) $A*B*CD+$

Answer: a

*Explanation: RPN is $AB*CD*+$.*

5. In computers, subtraction is generally carried out by _____

- a) 9's complement
- b) 10's complement
- c) 1's complement
- d) 2's complement

Answer: d

Explanation: Subtraction is done by 2's complement.

6. Assembly language _____

- a) Uses alphabetic codes in place of binary numbers used in machine language
- b) Is the easiest language to write programs

- c) Need not be translated into machine language
- d) None of the mentioned

Answer: a

Explanation: Uses binary numbers in machine language.

1. Choose the correct statement.

- a) CFG is not LR
- b) Ambiguous Grammar can never be LR
- c) CFG is not LR & Ambiguous Grammar can never be LR
- d) None of the mentioned

Answer: c

Explanation: Mentioned reason is true.

2. How is the parsing precedence relations defined?

- a) None of the mentioned
- b) All of the mentioned
- c) To delimit the handle
- d) Only for a certain pair of terminals

Answer: b

Explanation: The reason why the precedence operations is performed.

3. When will the relationship between '+' and '-' be <?

- a) For unary minus
- b) Minus is right associative
- c) All of the mentioned
- d) None of the mentioned

Answer: c

Explanation: Both statements are true.

4. When will the relationship between '<' and '>' be <?

- a) >
- b) <
- c) =
- d) Undefined

Answer: d

Explanation: Undefined. There is no existing relationship between the two.

1. The grammar $S \rightarrow aSa \mid bS \mid c$ is?

- a) LL(1) but not LR(1)
- b) LR(1) but not LR(1)
- c) Both LL(1) but not LR(1) & LR(1) but not LR(1)
- d) None of the mentioned

Answer: c

Explanation:

$First(aSa) = a$

$First(bS) = b$

$First(c) = c$

LR parsers are more powerful than LL (1) parsers and LR (1).

2. Recursive descent parsing is an example of _____

- a) Top down parsing
- b) Bottom up parsing

- c) Predictive parsing**
- d) None of the mentioned**

Answer: a

Explanation: Top down is the answer.

3. LR stands for _____

- a) Left to right**
- b) Left to right reduction**
- c) Right to left**
- d) Right most derivation and Left to right and a in reverse**

Answer: d

Explanation: Right most derivation and left to right and in reverse is used for LR.

4. Which is the most powerful parser?

- a) SLR**
- b) LALR**
- c) Canonical LR**
- d) Operator-precedence**

Answer: c

Explanation: Canonical tops all other parsers.