Unit 1

ALGORITHMIC PROBLEM SOLVING

ANNA UNIVERSITY 2 MARK QUESTIONS WITH ANSWERS

1. What is an algorithm? **(AU Jan 2018, R2017)**
(AU Nov/Dec 2015, Nov/Dec 2014, May/June 2013, Jan 2013, Jan 2012, May/June 2012, Jan 2010)

Algorithm is an ordered sequence of finite, well defined, unambiguous instructions for completing a task.

2. Write an algorithm to accept two numbers. Compute sum and print the result. **(AU Jan 2018, R2017)**

```
1. Start
2. Read two numbers A and B from user
3. Assign Sum=0
4. Calculate Sum = A + B
5. Print sum
6. Stop
```

3. What is a pseudocode? **(AU Nov/Dec 2015, Jan 2013, May/June 2009)**

Pseudocode is a of short, readable and formally styled English language instructions used for explaining an algorithm.

4. What are flowcharts and list down their advantages?
(AU Nov/Dec 2014, May/June 2013, May/June 2012)

A flowchart is a diagrammatic representation of the logic for solving a task.

- A flowchart helps to clarify how things are currently working and how they are improved.

- It makes communication among participants and establishes a common understanding them about the process.
- It helps to remove repeated and misplaced steps.
- It also helps in monitoring important areas, data collection and identifies areas for improvement or increase in efficiency.

5. Enlist the advantages of algorithm. (AU Jan 2012)

1. It is a simple to understand step by step solution of the problem.
2. It is easy to debug.
3. An algorithm is independent of programming languages.

6. What is the difference between program and algorithm? (AU April/May 2011)

Program: A program is a set of instruction given to a computer to solve a problem.

Algorithm: An algorithm is a procedure or finite sequence of steps that provides the solution to a problem.

7. What are the benefits of pseudocode. (AU Apr/May 2011)

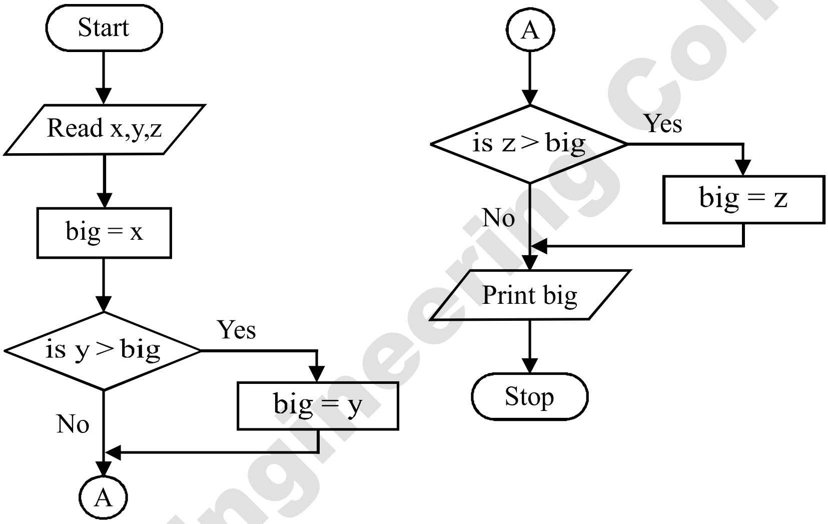
- Pseudocode is language independent code. It can be used by most programmers.
- It is easy to develop a program from pseudocode. Programmers can focus on the steps to solve a problem rather than on how to use the computer language.
- It is easy to translate pseudocode into a program in a high level programming language and it can even be achieved by less experienced programmers.
- The use of words and phrases in pseudocode are in line with basic computer operations, simplifying the translation from the pseudocode to a specific programming language.
- Pseudocode is compact, more readable and more easy to modify.

8. Write down steps involved in writing a program to solve a problem. (AU May/June 2010)

To design a program, a programmer must determine three basic steps.

1. The instructions to be performed.
2. The sequence in which those instructions are to be performed.
3. The data required to perform those instructions.

9. Draw a flowchart to find the maximum among the three numbers. (AU Jan 2009)



10. List the different building blocks of algorithms.

The algorithms can be constructed from basic building blocks. These building blocks are,

- Instructions/statements
- State
- Control flow
- Functions

11. What is coding?

Translating an algorithm into a programming language is called coding the algorithm.

12. Define Iteration.

The iteration is a technique that repeats the execution of a block of instructions.

13. Define recursion.

A function or procedure is called by itself again and again until a given condition is satisfied, is called recursion.

ANNA UNIVERSITY 16 MARK QUESTIONS WITH ANSWERS

1. Draw a flowchart to accept three distinct numbers find the greatest and print the result. (8)

   (AU Jan 2018, R2017)

   Refer Page No.: 47

2. Draw a flowchart to find the sum of the series 1+ 2 + 3 + 4 + 5 +……+ 100. (8)

   (AU Jan 2018, R2017)

   Refer Page No.: 15

3. Outline the Towers of Hanoi problem. Suggest a solution to the Towers of Hanoi problem with relevant diagrams. (16)

   (AU Jan 2018, R2017)

   Refer Page No.: 33

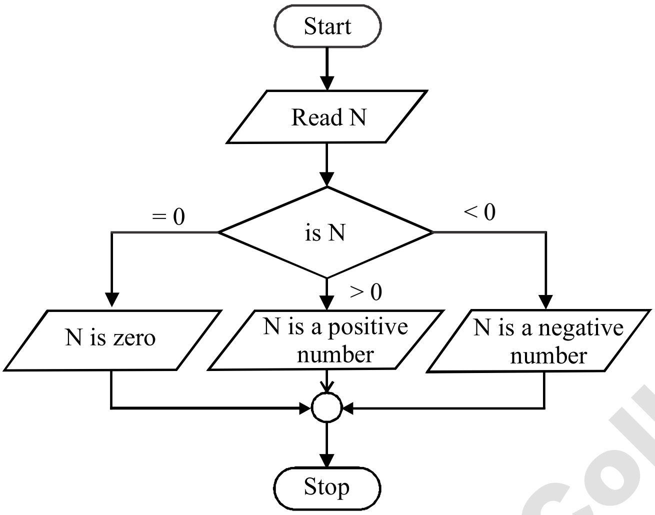
4. Explain about algorithm, pseudocode and flowchart with an example of finding the sum of 'n' numbers. (16)

   (AU Nov/Dec 2016)

   Refer Page No.: 48

5. Draw a flowchart to check whether the given number is zero, positive or negative. (6)

   (AU April/May 2015)



6. What is pseudocode? Write a pseudocode for swapping two numbers without using temporary storage. (6)

(AU Nov/Dec 2014)

Refer Page No.: 6

Pseudocode for swapping two numbers

```
1. READ A, B,
2. CALCULATE
   B = A + B
   A = B – A
   B = B – A
3. PRINT A, B
```

7. Discuss the need for logical analysis with an example in brief. (4)

(AU Nov/Dec 2014)

Refer Page No.: 1

8. Draw a flowchart to find the factorial of a number. (6)

(AU May/June 2014)

Refer Page No.: 52

9. Draw a flowchart to find the sum of first 100 natural numbers. (6)

(AU May/June 2014)

Refer Page No.: 48

10. Write an algorithm to find the largest of three numbers. (8)

(AU Jan 2013, May/ June 2010)

Refer Page No.: 2

11. Draw the flowchart for finding the roots of a quadratic equation. (8)

(AU May/June 2016, Jan 2013)

Refer Page No.: 45

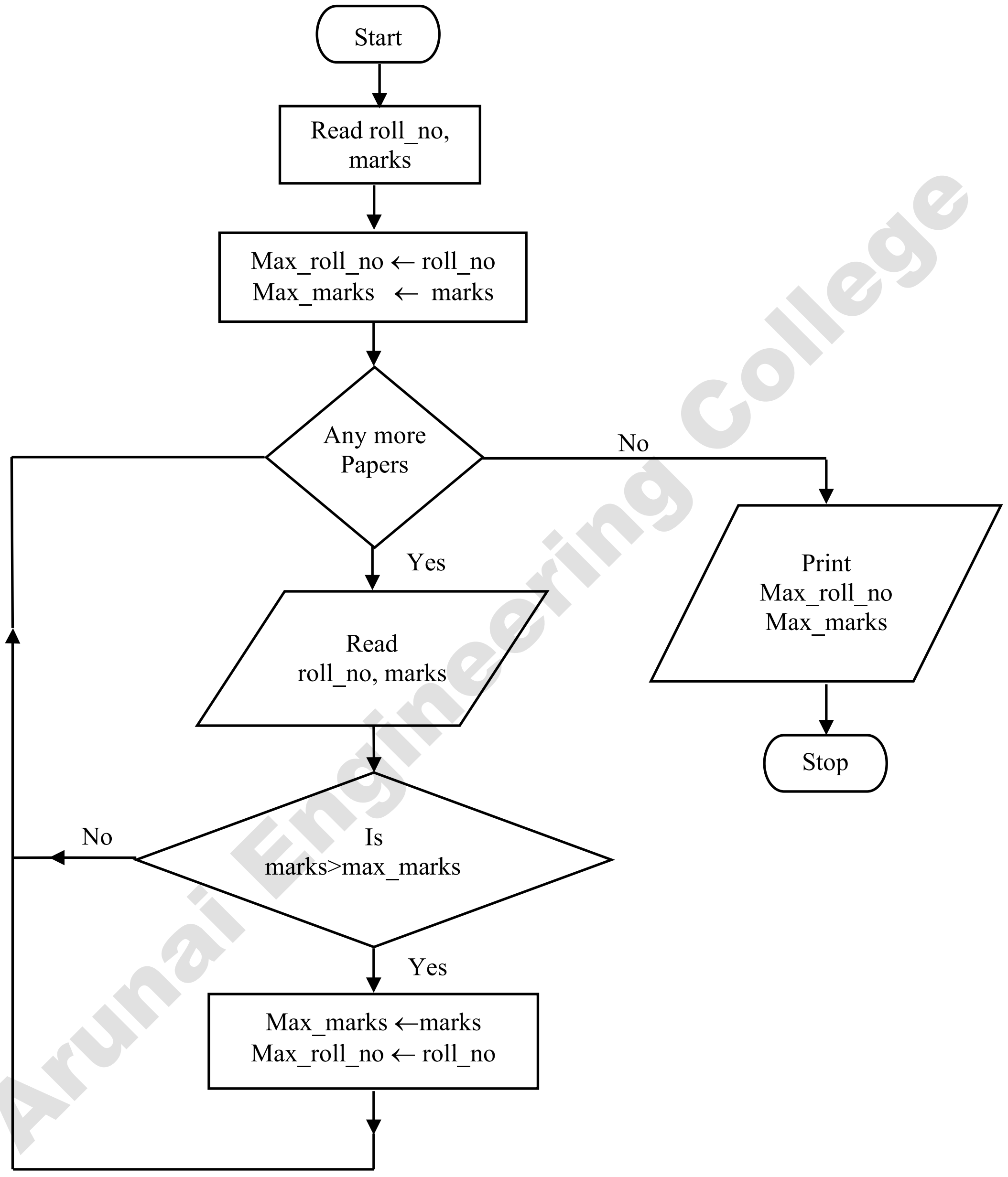
12. What is pseudocode? How does it differ from flowchart? Write a pseudocode to add up all the even numbers between 0 and 100 and print the result. (8)

(AU May/June 2016, May/June 2012, May/June 2010)

Refer Page No.: 6 & 7

13. Draw a flowchart to find the highest mark in a set of n marks. (8)

(AU Jan 2012)



14. Explain the guidelines for preparing flowcharts, benefits and limitation of flowcharts. (8)

(AU May/June 2012)

Refer Page No.: 12

15. Write an algorithm for finding sum and average of n numbers. Also, state the properties of a good algorithm. (8)

(AU May/June 2012)

Refer Page No.: 3 & 37

16. Write an algorithm to compute the factorial of a number n. (8)

(AU Jan 2012)

1. Start the program.
2. Read a number.
3. Set a loop to find the factorial of the given number using the formula fact = fact * i.
4. Print the factorial of the given number.
5. Stop the program.

17. Write a pseudocode to add two matrices. (8)

(AU Jan 2012)

Refer Page No.: 54

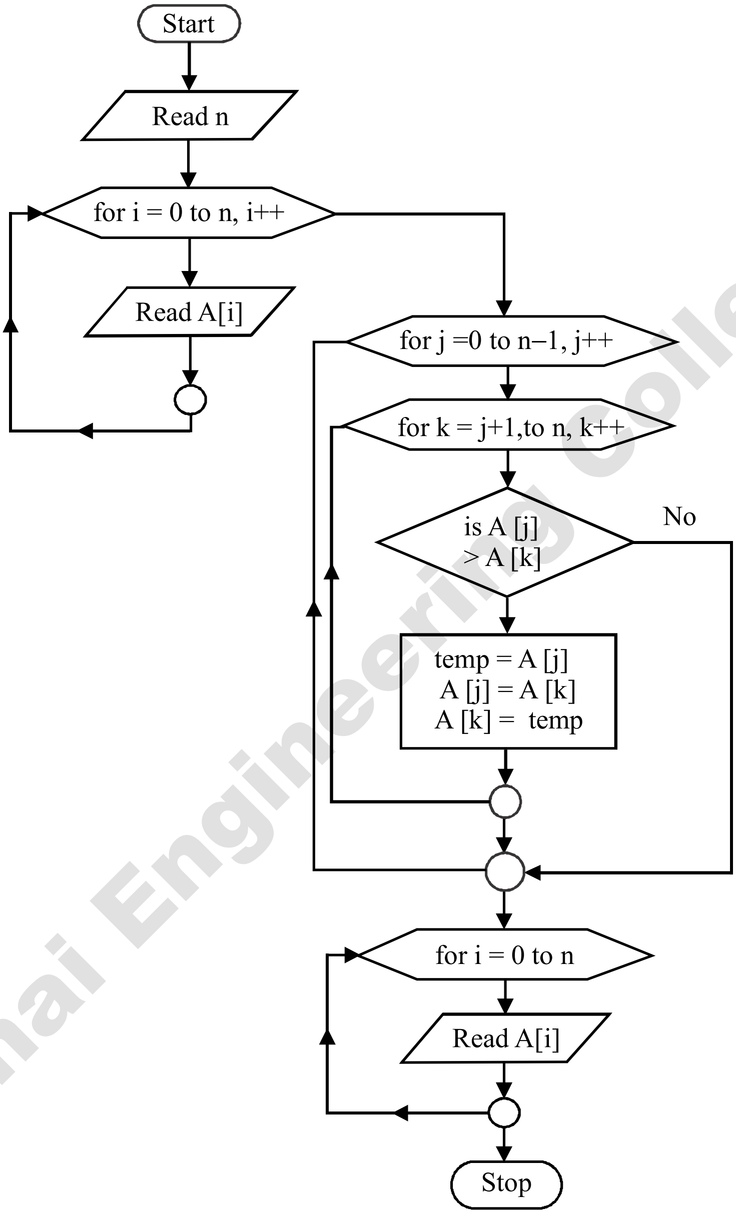
18. Draw the flowchart and write an algorithm to sort a list of numbers in ascending order. (16)

(AU April/May 2011)

Algorithm:

1. Start.
2. Read the size of the array.
3. Read the element of the array
4. Set a loop up to the array size minus one.
5. Set a inner loop up to the size of the array.
6. Check whether the adjacent array element is greater than or not.
7. If greater than exchange their position.
8. If not greater than then go to the loop.
9. After the execution of the inner loop, the outer loop is executed.
10. Print the ascending order of the given array.
11. Stop.

Flowchart:



19. Draw and explain the various symbols of flowchart and also draw the flowchart to add an array of N elements.

(AU Jan 2011, Jan 2010)

Refer Page No.: 11

20. Draw a flowchart to add two matrices.

(AU Jan 2009, May/June 2009)

Refer Page No.: 54

21. Write an algorithm to print the Fibonacci series (0, 1, 1, 2, 5, 8, 13,…..).

(AUMay/June 2009)

1. Start.
2. Read num.
3. Assign f1 = 0, f2 = 1.
4. Set loop for i and for all i < num value and calculate f3 = f1+ f2;
5. Display f1.
6. Assign f1= f2 and f2 = f3.
7. Stop.

22. Write a pseudocode for calculating compound interest.

(AU May/June 2009)

```
1. READ P, R, N, T
2. CALCULATE
   CI = P *(I + R/N) * N * T
3. PRINT CT
```

23. What is an algorithm? Write an algorithm to print numbers from 2 to 100. (8)

(AU Jan 2009)

Refer Page No.: 2

1. Start
2. Using loop generate i from 2 to 100.
3. Print i
4. Stop.

24. Write a pseudocode to multiply two matrices (8)

(AU Jan 2009)

```
READ A[i] [j], B[i][j]
FOR (i = 0, i < 3; i++)
        FOR (j = 0;  j < 3;  j++)
               FOR (k=0;  k<3; k++)
               CALCULATE C[i][j] = C[i][j] + A[i][k] * B[k][j]
```

```
            END FOR
        END FOR
    END FOR
    PRINT C[i][j]
```

25. Explain the building blocks of algorithms in detail?

    Refer Page No.: 3 to 6

26. Explain the steps in algorithmic problem solving techniques in detail?

    Refer Page No.: 17

27. Explain different strategies for developing algorithms?

    Refer Page No.: 20