

Accenture Technical Assessment 2026 Batch By - Mr. Durgesh StudyHub

Pseudocode / Logic– 200- Questions (With Options & Answers)

Q1. Find the sum of two numbers.

Pseudocode:

```
START
  INPUT A, B
  SUM = A + B
  OUTPUT SUM
END
```

Options:

A) $A + B$ ☒

B) $A - B$

C) $A * B$

D) A / B

Answer: A) $A + B$

Q2. Find the factorial of a number N.

Pseudocode:

START

 INPUT N

 FACT = 1

 FOR I = 1 TO N

 FACT = FACT * I

 END FOR

 OUTPUT FACT

END

Options:

A) Product of 1 to N ☒

B) Sum of 1 to N

C) N squared

D) N divided by 2

Answer: A) Product of 1 to N

Q3. Find the largest of two numbers A and B.

Pseudocode:

```
START
  INPUT A, B
  IF A > B THEN
    OUTPUT A
  ELSE
    OUTPUT B
  ENDIF
END
```

Options:

A) Smaller number

B) Larger number ☒

C) Sum of A and B

D) Product of A and B

Answer: B) Larger number

Q4. Check if a number is even or odd.

Pseudocode:

```
START
  INPUT N
  IF N MOD 2 = 0 THEN
    OUTPUT "Even"
  ELSE
    OUTPUT "Odd"
  ENDIF
END
```

Options:

- A) Even or Odd ☒
- B) Prime or Composite
- C) Positive or Negative
- D) Zero

Answer: A) Even or Odd

Q5. Find the sum of first N natural numbers.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    SUM = SUM + I
  END FOR
  OUTPUT SUM
END
```

Options:

- A) $N!$
- B) N squared
- C) Sum of 1 to N ☒
- D) $N / 2$

Answer: C) Sum of 1 to N

Q6. Print all numbers from 1 to N .

Pseudocode:

```
START
  INPUT N
```

```
FOR I = 1 TO N
    OUTPUT I
END FOR
END
```

Options:

- A) Numbers 1 to N ☒
- B) Numbers 1 to N squared
- C) Numbers N to 1
- D) Factorial of N

Answer: A) Numbers 1 to N

Q7. Find the reverse of a number.

Pseudocode:

```
START
    INPUT N
    REV = 0
    WHILE N > 0
        DIGIT = N MOD 10
        REV = REV * 10 + DIGIT
        N = N / 10
    END WHILE
```

```
    OUTPUT REV
END
```

Options:

A) Reverse of N ☒

B) N squared

C) Sum of digits

D) Product of digits

Answer: A) Reverse of N

Q8. Find GCD of two numbers.

Pseudocode:

```
START
    INPUT A, B
    WHILE B != 0
        TEMP = B
        B = A MOD B
        A = TEMP
    END WHILE
    OUTPUT A
END
```

Options:

A) LCM

B) GCD ☒

C) Sum

D) Difference

Answer: B) GCD

Q9. Check if a number is prime.

Pseudocode:

START

 INPUT N

 FLAG = 0

 FOR I = 2 TO N-1

 IF N MOD I = 0 THEN

 FLAG = 1

 BREAK

 ENDIF

 END FOR

 IF FLAG = 0 THEN

 OUTPUT "Prime"

 ELSE

```
        OUTPUT "Not Prime"
    ENDIF
END
```

Options:

- A) Prime or Not Prime ☒
- B) Even or Odd
- C) Positive or Negative
- D) Multiple of 10

Answer: A) Prime or Not Prime

Q10. Find sum of digits of a number.

Pseudocode:

```
START
    INPUT N
    SUM = 0
    WHILE N > 0
        DIGIT = N MOD 10
        SUM = SUM + DIGIT
        N = N / 10
    END WHILE
    OUTPUT SUM
```

END

Options:

A) Product of digits

B) Sum of digits ☒

C) Reverse number

D) Factorial

Answer: B) Sum of digits

Q11. Find the product of first N natural numbers.

Pseudocode:

START

 INPUT N

 PRODUCT = 1

 FOR I = 1 TO N

 PRODUCT = PRODUCT * I

 END FOR

 OUTPUT PRODUCT

END

Options:

A) $N!$ ☒

B) Sum of 1 to N

C) N squared

D) $N / 2$

Answer: A) $N!$

Q12. Find the smallest of three numbers A, B, C.

Pseudocode:

START

INPUT A, B, C

MIN = A

IF B < MIN THEN

MIN = B

ENDIF

IF C < MIN THEN

MIN = C

ENDIF

OUTPUT MIN

END

Options:

A) Largest number

B) Smallest number ☒

C) Sum of numbers

D) Product of numbers

Answer: B) Smallest number

Q13. Find the largest of three numbers A, B, C.

Pseudocode:

START

INPUT A, B, C

MAX = A

IF B > MAX THEN

MAX = B

ENDIF

IF C > MAX THEN

MAX = C

ENDIF

```
    OUTPUT MAX
END
```

Options:

- A) Smallest number
- B) Largest number ☒
- C) Sum of numbers
- D) Product of numbers

Answer: B) Largest number

Q14. Print all even numbers from 1 to N.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    IF I MOD 2 = 0 THEN
      OUTPUT I
    ENDIF
  END FOR
END
```

Options:

A) Odd numbers

B) Even numbers ☒

C) Prime numbers

D) Fibonacci numbers

Answer: B) Even numbers

Q15. Print all odd numbers from 1 to N.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 IF I MOD 2 \neq 0 THEN

 OUTPUT I

 ENDIF

 END FOR

END

Options:

A) Even numbers

B) Odd numbers ☒

- C) Prime numbers
- D) Fibonacci numbers

Answer: B) Odd numbers

Q16. Find sum of squares of first N natural numbers.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    SUM = SUM + I*I
  END FOR
  OUTPUT SUM
END
```

Options:

- A) Sum of numbers
- B) Sum of squares ☒
- C) Sum of cubes

D) Product of numbers

Answer: B) Sum of squares

Q17. Check if a number is positive, negative, or zero.

Pseudocode:

```
START
  INPUT N
  IF N > 0 THEN
    OUTPUT "Positive"
  ELSE IF N < 0 THEN
    OUTPUT "Negative"
  ELSE
    OUTPUT "Zero"
  ENDIF
END
```

Options:

A) Positive, Negative, or Zero ☒

B) Even or Odd

C) Prime or Not Prime

D) Multiple of 10

Answer: A) Positive, Negative, or Zero

Q18. Find the average of N numbers.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    INPUT NUM
    SUM = SUM + NUM
  END FOR
  AVERAGE = SUM / N
  OUTPUT AVERAGE
END
```

Options:

A) Maximum

B) Minimum

C) Average ☒

D) Sum

Answer: C) Average

Q19. Print Fibonacci series up to N terms.

Pseudocode:

```
START
  INPUT N
  A = 0
  B = 1
  OUTPUT A, B
  FOR I = 3 TO N
    C = A + B
    OUTPUT C
    A = B
    B = C
  END FOR
END
```

Options:

A) Fibonacci series ☒

- B) Prime series
- C) Even numbers
- D) Odd numbers

Answer: A) Fibonacci series

Q20. Find the sum of even numbers from 1 to N.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    IF I MOD 2 = 0 THEN
      SUM = SUM + I
    ENDIF
  END FOR
  OUTPUT SUM
END
```

Options:

- A) Sum of odd numbers

B) Sum of even numbers ☒

C) Sum of all numbers

D) Product of numbers

Answer: B) Sum of even numbers

Q21. Find the sum of odd numbers from 1 to N.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    IF I MOD 2 != 0 THEN
      SUM = SUM + I
    ENDIF
  END FOR
  OUTPUT SUM
END
```

Options:

A) Sum of even numbers

B) Sum of odd numbers ☒

C) Product of numbers

D) Average of numbers

Answer: B) Sum of odd numbers

Q22. Check if a number is palindrome.

Pseudocode:

START

 INPUT N

 TEMP = N

 REV = 0

 WHILE N > 0

 DIGIT = N MOD 10

 REV = REV * 10 + DIGIT

 N = N / 10

 END WHILE

 IF TEMP = REV THEN

 OUTPUT "Palindrome"

 ELSE

 OUTPUT "Not Palindrome"

 ENDIF

END

Options:

A) Palindrome or Not ☒

B) Even or Odd

C) Positive or Negative

D) Prime or Not Prime

Answer: A) Palindrome or Not

Q23. Find LCM of two numbers.

Pseudocode:

START

 INPUT A, B

 LCM = MAX (A, B)

 WHILE LCM MOD A \neq 0 OR
LCM MOD B \neq 0

 LCM = LCM + 1

 END WHILE

 OUTPUT LCM

END

Options:

A) LCM ☒

B) GCD

C) Sum

D) Product

Answer: A) LCM

Q24. Print multiplication table of a number N.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO 10
    PRINT N * I
  END FOR
END
```

Options:

A) Addition table

B) Multiplication table ☒

C) Subtraction table

D) Division table

Answer: B) Multiplication table

Q25. Find sum of digits of a number.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  WHILE N > 0
    DIGIT = N MOD 10
    SUM = SUM + DIGIT
    N = N / 10
  END WHILE
  OUTPUT SUM
END
```

Options:

A) Product of digits

B) Sum of digits ☒

C) Reverse number

D) Factorial

Answer: B) Sum of digits

Q26. Find product of digits of a number.

Pseudocode:

```
START
    INPUT N
    PRODUCT = 1
    WHILE N > 0
        DIGIT = N MOD 10
        PRODUCT = PRODUCT *
DIGIT
        N = N / 10
    END WHILE
    OUTPUT PRODUCT
END
```

Options:

A) Product of digits ☒

B) Sum of digits

C) Reverse number

D) Factorial

Answer: A) Product of digits

Q27. Count the number of digits in a number.

Pseudocode:

```
START
  INPUT N
  COUNT = 0
  WHILE N > 0
    N = N / 10
    COUNT = COUNT + 1
  END WHILE
  OUTPUT COUNT
END
```

Options:

A) Number of digits ☒

B) Sum of digits

C) Product of digits

D) Factorial

Answer: A) Number of digits

Q28. Check if a number is Armstrong.

Pseudocode:

START

 INPUT N

 TEMP = N

 SUM = 0

 WHILE N > 0

 DIGIT = N MOD 10

 SUM = SUM + DIGIT³

 N = N / 10

 END WHILE

 IF TEMP = SUM THEN

 OUTPUT "Armstrong"

 ELSE

 OUTPUT "Not Armstrong"

 ENDIF

END

Options:

A) Armstrong or Not ☒

B) Even or Odd

C) Prime or Not

D) Positive or Negative

Answer: A) Armstrong or Not

Q29. Find reverse of a number.

Pseudocode:

START

 INPUT N

 REV = 0

 WHILE N > 0

 DIGIT = N MOD 10

 REV = REV * 10 + DIGIT

 N = N / 10

 END WHILE

 OUTPUT REV

END

Options:

A) Reverse of number ☒

B) Sum of digits

C) Product of digits

D) Factorial

Answer: A) Reverse of number

Q30. Find sum of first N even numbers.

Pseudocode:

START

 INPUT N

 SUM = 0

 FOR I = 1 TO N

 SUM = SUM + (2 * I)

 END FOR

 OUTPUT SUM

END

Options:

A) Sum of first N odd numbers

B) Sum of first N even numbers ☒

C) Sum of first N natural numbers

D) Product of first N numbers

Answer: B) Sum of first N even numbers

Q31. Find sum of first N odd numbers.

Pseudocode:

START

 INPUT N

 SUM = 0

 FOR I = 1 TO N

 SUM = SUM + (2*I - 1)

 END FOR

 OUTPUT SUM

END

Options:

A) Sum of first N odd numbers ☒

B) Sum of first N even numbers

C) Sum of numbers from 1 to N

D) Product of numbers

Answer: A) Sum of first N odd numbers

Q32. Check if a number is prime.

Pseudocode:

```
START
  INPUT N
  FLAG = 0
  FOR I = 2 TO N-1
    IF N MOD I = 0 THEN
      FLAG = 1
      BREAK
    ENDIF
  END FOR
  IF FLAG = 0 THEN
    OUTPUT "Prime"
  ELSE
    OUTPUT "Not Prime"
  ENDIF
END
```

Options:

- A) Prime or Not ☒
- B) Even or Odd

C) Positive or Negative

D) Armstrong or Not

Answer: A) Prime or Not

Q33. Print all prime numbers up to N.

Pseudocode:

START

 INPUT N

 FOR I = 2 TO N

 FLAG = 0

 FOR J = 2 TO I-1

 IF I MOD J = 0 THEN

 FLAG = 1

 BREAK

 ENDIF

 END FOR

 IF FLAG = 0 THEN

 OUTPUT I

 ENDIF

 END FOR

END

Options:

A) Prime numbers ☒

B) Even numbers

C) Odd numbers

D) Fibonacci numbers

Answer: A) Prime numbers

Q34. Find sum of prime numbers up to N.

Pseudocode:

START

 INPUT N

 SUM = 0

 FOR I = 2 TO N

 FLAG = 0

 FOR J = 2 TO I-1

 IF I MOD J = 0 THEN

 FLAG = 1

 BREAK

 ENDIF

 END FOR

```
    IF FLAG = 0 THEN
        SUM = SUM + I
    ENDIF
END FOR
OUTPUT SUM
END
```

Options:

- A) Sum of prime numbers ☒
- B) Sum of even numbers
- C) Sum of odd numbers
- D) Sum of digits

Answer: A) Sum of prime numbers

Q35. Print first N terms of Fibonacci series.

Pseudocode:

```
START
    INPUT N
    A = 0
    B = 1
    OUTPUT A, B
```

```
FOR I = 3 TO N
    C = A + B
    OUTPUT C
    A = B
    B = C
END FOR
END
```

Options:

- A) Fibonacci series ☒
- B) Prime series
- C) Even numbers
- D) Odd numbers

Answer: A) Fibonacci series

Q36. Find factorial of a number.

Pseudocode:

```
START
    INPUT N
    FACT = 1
    FOR I = 1 TO N
        FACT = FACT * I
```

```
END FOR
  OUTPUT FACT
END
```

Options:

- A) Factorial ☒
- B) Sum
- C) Product of digits
- D) Sum of squares

Answer: A) Factorial

Q37. Find GCD of two numbers.

Pseudocode:

```
START
  INPUT A, B
  WHILE B != 0
    TEMP = B
    B = A MOD B
    A = TEMP
  END WHILE
  OUTPUT A
END
```

Options:

A) GCD ☒

B) LCM

C) Sum

D) Product

Answer: A) GCD

Q38. Find LCM of two numbers.

Pseudocode:

START

 INPUT A, B

 LCM = MAX (A, B)

 WHILE LCM MOD A \neq 0 OR
LCM MOD B \neq 0

 LCM = LCM + 1

 END WHILE

 OUTPUT LCM

END

Options:

A) LCM ☒

B) GCD

C) Sum

D) Product

Answer: A) LCM

Q39. Check if a number is palindrome.

Pseudocode:

START

 INPUT N

 TEMP = N

 REV = 0

 WHILE N > 0

 DIGIT = N MOD 10

 REV = REV * 10 + DIGIT

 N = N / 10

 END WHILE

 IF TEMP = REV THEN

 OUTPUT "Palindrome"

 ELSE

 OUTPUT "Not Palindrome"

 ENDIF

END

Options:

A) Palindrome or Not ☒

B) Even or Odd

C) Positive or Negative

D) Prime or Not

Answer: A) Palindrome or Not

Q40. Find sum of digits of a number.

Pseudocode:

START

 INPUT N

 SUM = 0

 WHILE N > 0

 DIGIT = N MOD 10

 SUM = SUM + DIGIT

 N = N / 10

 END WHILE

 OUTPUT SUM

END

Options:

A) Sum of digits ☒

B) Product of digits

C) Reverse number

D) Factorial

Answer: A) Sum of digits

Q41. Find product of digits of a number.

Pseudocode:

START

 INPUT N

 PRODUCT = 1

 WHILE N > 0

 DIGIT = N MOD 10

 PRODUCT = PRODUCT * DIGIT

 DIGIT

 N = N / 10

 END WHILE

 OUTPUT PRODUCT

END

Options:

A) Product of digits ☒

B) Sum of digits

C) Reverse number

D) Factorial

Answer: A) Product of digits

Q42. Count digits in a number.

Pseudocode:

START

 INPUT N

 COUNT = 0

 WHILE N > 0

 N = N / 10

 COUNT = COUNT + 1

 END WHILE

 OUTPUT COUNT

END

Options:

A) Count digits ☒

- B) Sum of digits
- C) Product of digits
- D) Factorial

Answer: A) Count digits

Q43. Check if number is Armstrong.

Pseudocode:

START

 INPUT N

 TEMP = N

 SUM = 0

 WHILE N > 0

 DIGIT = N MOD 10

 SUM = SUM + DIGIT³

 N = N / 10

 END WHILE

 IF TEMP = SUM THEN

 OUTPUT "Armstrong"

 ELSE

 OUTPUT "Not Armstrong"

 ENDIF

END

Options:

A) Armstrong or Not ☒

B) Even or Odd

C) Prime or Not

D) Positive or Negative

Answer: A) Armstrong or Not

Q44. Print first N even numbers.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 OUTPUT 2 * I

 END FOR

END

Options:

A) First N even numbers ☒

B) First N odd numbers

C) First N prime numbers

D) First N natural numbers

Answer: A) First N even numbers

Q45. Print first N odd numbers.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    OUTPUT 2*I - 1
  END FOR
END
```

Options:

A) First N even numbers

B) First N odd numbers ☒

C) First N prime numbers

D) First N natural numbers

Answer: B) First N odd numbers

Q46. Find sum of first N even numbers.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    SUM = SUM + 2*I
  END FOR
  OUTPUT SUM
END
```

Options:

- A) Sum of first N even numbers ☒
- B) Sum of first N odd numbers
- C) Sum of first N natural numbers
- D) Product of first N numbers

Answer: A) Sum of first N even numbers

Q47. Find sum of first N odd numbers.

Pseudocode:

```
START
  INPUT N
```

```
SUM = 0
FOR I = 1 TO N
    SUM = SUM + (2*I - 1)
END FOR
OUTPUT SUM
END
```

Options:

- A) Sum of first N odd numbers ☒
- B) Sum of first N even numbers
- C) Sum of numbers from 1 to N
- D) Product of numbers

Answer: A) Sum of first N odd numbers

Q48. Find factorial of a number.

Pseudocode:

```
START
    INPUT N
    FACT = 1
    FOR I = 1 TO N
        FACT = FACT * I
    END FOR
```

```
    OUTPUT FACT
END
```

Options:

- A) Factorial ☒
- B) Sum
- C) Product of digits
- D) Sum of squares

Answer: A) Factorial

Q49. Check if number is prime.

Pseudocode:

```
START
  INPUT N
  FLAG = 0
  FOR I = 2 TO N-1
    IF N MOD I = 0 THEN
      FLAG = 1
      BREAK
    ENDIF
  END FOR
  IF FLAG = 0 THEN
```

```
        OUTPUT "Prime"
ELSE
        OUTPUT "Not Prime"
ENDIF
END
```

Options:

- A) Prime or Not ☒
- B) Even or Odd
- C) Positive or Negative
- D) Armstrong or Not

Answer: A) Prime or Not

Q50. Print all prime numbers up to N.

Pseudocode:

```
START
  INPUT N
  FOR I = 2 TO N
    FLAG = 0
    FOR J = 2 TO I-1
      IF I MOD J = 0 THEN
        FLAG = 1
```

```
        BREAK
    ENDIF
END FOR
IF FLAG = 0 THEN
    OUTPUT I
ENDIF
END FOR
END
```

Options:

- A) Prime numbers ☒
- B) Even numbers
- C) Odd numbers
- D) Fibonacci numbers

Answer: A) Prime numbers

Q51. Find the sum of first N natural numbers divisible by 3.

Pseudocode:

START

```
INPUT N
SUM = 0
FOR I = 1 TO N
    IF I MOD 3 = 0 THEN
        SUM = SUM + I
    ENDIF
END FOR
OUTPUT SUM
END
```

Options:

- A) Sum of numbers divisible by 3 ☒
- B) Sum of numbers divisible by 2
- C) Sum of all numbers
- D) Product of numbers

Answer: A) Sum of numbers divisible by 3

Q52. Find the sum of first N natural numbers divisible by 5.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    IF I MOD 5 = 0 THEN
      SUM = SUM + I
    ENDIF
  END FOR
  OUTPUT SUM
END
```

Options:

- A) Sum of numbers divisible by 5 ☒
- B) Sum of numbers divisible by 3
- C) Sum of all numbers
- D) Product of numbers

Answer: A) Sum of numbers divisible by 5

Q53. Print all numbers divisible by 7 up to N.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    IF I MOD 7 = 0 THEN
      OUTPUT I
    ENDIF
  END FOR
END
```

Options:

- A) Numbers divisible by 7 ☒
- B) Numbers divisible by 3
- C) Prime numbers
- D) Fibonacci numbers

Answer: A) Numbers divisible by 7

Q54. Count numbers divisible by 4 up to N.

Pseudocode:

```
START
  INPUT N
  COUNT = 0
```

```
FOR I = 1 TO N
    IF I MOD 4 = 0 THEN
        COUNT = COUNT + 1
    ENDIF
END FOR
OUTPUT COUNT
END
```

Options:

- A) Count of numbers divisible by 4 ☒
- B) Count of prime numbers
- C) Count of odd numbers
- D) Count of even numbers

Answer: A) Count of numbers divisible by 4

Q55. Print first N terms of geometric series with first term A and ratio R.

Pseudocode:

```
START
    INPUT N, A, R
```

```
TERM = A
FOR I = 1 TO N
    OUTPUT TERM
    TERM = TERM * R
END FOR
END
```

Options:

- A) Geometric series ☒
- B) Arithmetic series
- C) Fibonacci series
- D) Prime numbers

Answer: A) Geometric series

Q56. Print first N terms of arithmetic series with first term A and difference D.

Pseudocode:

```
START
    INPUT N, A, D
    TERM = A
    FOR I = 1 TO N
        OUTPUT TERM
```

```
        TERM = TERM + D
    END FOR
END
```

Options:

- A) Arithmetic series ☒
- B) Geometric series
- C) Fibonacci series
- D) Prime numbers

Answer: A) Arithmetic series

Q57. Find sum of first N terms of arithmetic series.

Pseudocode:

```
START
    INPUT N, A, D
    SUM = 0
    TERM = A
    FOR I = 1 TO N
        SUM = SUM + TERM
        TERM = TERM + D
    END FOR
```

```
    OUTPUT SUM
END
```

Options:

- A) Sum of arithmetic series ☒
- B) Sum of geometric series
- C) Sum of Fibonacci numbers
- D) Sum of prime numbers

Answer: A) Sum of arithmetic series

Q58. Find sum of first N terms of geometric series.

Pseudocode:

```
START
    INPUT N, A, R
    SUM = 0
    TERM = A
    FOR I = 1 TO N
        SUM = SUM + TERM
        TERM = TERM * R
    END FOR
    OUTPUT SUM
```

END

Options:

A) Sum of geometric series ☒

B) Sum of arithmetic series

C) Sum of Fibonacci numbers

D) Sum of prime numbers

Answer: A) Sum of geometric series

Q59. Check if a number is perfect number.

Pseudocode:

START

 INPUT N

 SUM = 0

 FOR I = 1 TO N-1

 IF N MOD I = 0 THEN

 SUM = SUM + I

 ENDIF

 END FOR

 IF SUM = N THEN

 OUTPUT "Perfect Number"

```
ELSE
    OUTPUT "Not Perfect
Number"
ENDIF
END
```

Options:

A) Perfect Number or Not ☒

B) Prime or Not

C) Even or Odd

D) Armstrong or Not

Answer: A) Perfect Number or Not

Q60. Print all perfect numbers up to N.

Pseudocode:

```
START
    INPUT N
    FOR NUM = 1 TO N
        SUM = 0
        FOR I = 1 TO NUM-1
            IF NUM MOD I = 0 THEN
                SUM = SUM + I
```

```
        ENDIF
    END FOR
    IF SUM = NUM THEN
        OUTPUT NUM
    ENDIF
END FOR
END
```

Options:

- A) Perfect numbers ☒
- B) Prime numbers
- C) Fibonacci numbers
- D) Even numbers

Answer: A) Perfect numbers

Q61. Find sum of all perfect numbers up to N.

Pseudocode:

```
START
    INPUT N
    SUM_TOTAL = 0
    FOR NUM = 1 TO N
```

```
SUM = 0
FOR I = 1 TO NUM-1
    IF NUM MOD I = 0 THEN
        SUM = SUM + I
    ENDIF
END FOR
IF SUM = NUM THEN
    SUM_TOTAL = SUM_TOTAL
+ NUM
ENDIF
END FOR
OUTPUT SUM_TOTAL
END
```

Options:

- A) Sum of perfect numbers ☒
- B) Sum of prime numbers
- C) Sum of Fibonacci numbers
- D) Sum of even numbers

Answer: A) Sum of perfect numbers

Q62. Find HCF (GCD) of three numbers.

Pseudocode:

```
START
  INPUT A, B, C
  WHILE B != 0
    TEMP = B
    B = A MOD B
    A = TEMP
  END WHILE
  GCD = A
  WHILE C != 0
    TEMP = C
    C = GCD MOD C
    GCD = TEMP
  END WHILE
  OUTPUT GCD
END
```

Options:

- A) GCD ☒
- B) LCM
- C) Sum

D) Product

Answer: A) GCD

Q63. Find LCM of three numbers.

Pseudocode:

```
START
    INPUT A, B, C
    LCM_AB = MAX(A, B)
    WHILE LCM_AB MOD A != 0 OR
LCM_AB MOD B != 0
        LCM_AB = LCM_AB + 1
    END WHILE
    LCM_ABC = MAX(LCM_AB, C)
    WHILE LCM_ABC MOD
LCM_AB != 0 OR LCM_ABC MOD
C != 0
        LCM_ABC = LCM_ABC + 1
    END WHILE
    OUTPUT LCM_ABC
END
```

Options:

A) LCM ☒

B) GCD

C) Sum

D) Product

Answer: A) LCM

Q64. Find the second largest number in an array.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 INPUT ARR[I]

 END FOR

 MAX1 = -INF

 MAX2 = -INF

 FOR I = 1 TO N

 IF ARR[I] > MAX1 THEN

 MAX2 = MAX1

```
        MAX1 = ARR[I]
    ELSE IF ARR[I] > MAX2
AND ARR[I] != MAX1 THEN
        MAX2 = ARR[I]
    ENDIF
END FOR
OUTPUT MAX2
END
```

Options:

- A) Smallest number
- B) Second largest number ☒
- C) Largest number
- D) Sum of array

Answer: B) Second largest number

Q65. Count number of prime numbers in an array.

Pseudocode:

```
START
    INPUT N
    COUNT = 0
```

```
FOR I = 1 TO N
    INPUT ARR[I]
    FLAG = 0
    FOR J = 2 TO ARR[I]-1
        IF ARR[I] MOD J = 0
THEN
            FLAG = 1
            BREAK
        ENDIF
    END FOR
    IF FLAG = 0 AND ARR[I] >
1 THEN
        COUNT = COUNT + 1
    ENDIF
END FOR
OUTPUT COUNT
END
```

Options:

A) Total elements

B) Number of primes ☒

C) Number of even numbers

D) Sum of numbers

Answer: B) Number of primes

Q66. Reverse an array.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  FOR I = N DOWNT0 1
    OUTPUT ARR[I]
  END FOR
END
```

Options:

A) Original array

B) Sorted array

C) Reversed array ☒

D) Sum of array

Answer: C) Reversed array

Q67. Find maximum sum of contiguous subarray (Kadane's algorithm).

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  MAX_SO_FAR = ARR[1]
  MAX_ENDING = ARR[1]
  FOR I = 2 TO N
    MAX_ENDING = MAX(ARR[I],
MAX_ENDING + ARR[I])
    MAX_SO_FAR =
MAX(MAX_SO_FAR, MAX_ENDING)
  END FOR
  OUTPUT MAX_SO_FAR
END
```

Options:

A) Minimum element

B) Maximum element

C) Maximum sum of contiguous subarray



D) Sum of array

Answer: C) Maximum sum of contiguous subarray

Q68. Find intersection of two arrays.

Pseudocode:

START

 INPUT N, M

 FOR I = 1 TO N

 INPUT ARR1[I]

 END FOR

 FOR I = 1 TO M

 INPUT ARR2[I]

 END FOR

 FOR I = 1 TO N

 FOR J = 1 TO M

 IF ARR1[I] = ARR2[J]

THEN

```
        OUTPUT ARR1[I]
    ENDIF
END FOR
END FOR
END
```

Options:

- A) Union of arrays
- B) Intersection of arrays ☒
- C) Difference of arrays
- D) Sum of arrays

Answer: B) Intersection of arrays

Q69. Find transpose of a matrix.

Pseudocode:

```
START
    INPUT ROWS, COLS
    FOR I = 1 TO ROWS
        FOR J = 1 TO COLS
            INPUT MATRIX[I][J]
        END FOR
    END FOR
```

```
FOR I = 1 TO COLS
  FOR J = 1 TO ROWS
    OUTPUT MATRIX[J] [I]
  END FOR
END FOR
END
```

Options:

- A) Original matrix
- B) Transpose of matrix ☒
- C) Inverse of matrix
- D) Sum of matrix

Answer: B) Transpose of matrix

Q70. Find all pairs in an array with given sum S.

Pseudocode:

```
START
  INPUT N, S
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
```

```
FOR I = 1 TO N-1
    FOR J = I+1 TO N
        IF ARR[I] + ARR[J] = S
THEN
            OUTPUT ARR[I], ARR[J]
        ENDIF
    END FOR
END FOR
END
```

Options:

- A) Pairs with sum S ☒
- B) Pairs with product S
- C) Sum of array
- D) Product of array

Answer: A) Pairs with sum S

Q71. Find first non-repeating element in an array.

Pseudocode:

```
START
    INPUT N
```

```
FOR I = 1 TO N
    INPUT ARR[I]
END FOR
FOR I = 1 TO N
    COUNT = 0
    FOR J = 1 TO N
        IF ARR[I] = ARR[J]
THEN
            COUNT = COUNT + 1
        ENDIF
    END FOR
    IF COUNT = 1 THEN
        OUTPUT ARR[I]
        BREAK
    ENDIF
END FOR
END
```

Options:

- A) First repeating element
- B) First non-repeating element ☒
- C) Maximum element

D) Minimum element

Answer: B) First non-repeating element

Q72. Merge two sorted arrays into one sorted array.

Pseudocode:

START

 INPUT N, M

 FOR I = 1 TO N

 INPUT ARR1[I]

 END FOR

 FOR I = 1 TO M

 INPUT ARR2[I]

 END FOR

 i = 1, j = 1

 WHILE i <= N AND j <= M

 IF ARR1[i] < ARR2[j]

 THEN

 OUTPUT ARR1[i]

 i = i + 1

 ELSE

```
        OUTPUT ARR2[j]
        j = j + 1
    ENDIF
END WHILE
WHILE i <= N
    OUTPUT ARR1[i]
    i = i + 1
END WHILE
WHILE j <= M
    OUTPUT ARR2[j]
    j = j + 1
END WHILE
END
```

Options:

- A) Merge arrays ☒
- B) Intersection arrays
- C) Difference arrays
- D) Reverse arrays

Answer: A) Merge arrays

Q73. Rotate an array by K positions.

Pseudocode:

```
START
  INPUT N, K
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  FOR I = 1 TO K
    TEMP = ARR[N]
    FOR J = N DOWNT0 2
      ARR[J] = ARR[J-1]
    END FOR
    ARR[1] = TEMP
  END FOR
  FOR I = 1 TO N
    OUTPUT ARR[I]
  END FOR
END
```

Options:

- A) Array rotated ☒
- B) Array reversed
- C) Array sorted

D) Array sum

Answer: A) Array rotated

Q74. Find missing number in an array of 1 to N.

Pseudocode:

```
START
  INPUT N
  SUM = N * (N + 1) / 2
  FOR I = 1 TO N - 1
    INPUT ARR[I]
    SUM = SUM - ARR[I]
  END FOR
  OUTPUT SUM
END
```

Options:

A) Missing number ☒

B) Maximum number

C) Minimum number

D) Sum of array

Answer: A) Missing number

Q75. Find maximum and minimum in an array.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 INPUT ARR[I]

 END FOR

 MAX = ARR[1]

 MIN = ARR[1]

 FOR I = 2 TO N

 IF ARR[I] > MAX THEN

 MAX = ARR[I]

 ENDIF

 IF ARR[I] < MIN THEN

 MIN = ARR[I]

```
        ENDIF
    END FOR
    OUTPUT MAX, MIN
END
```

Options:

- A) Maximum and Minimum ☒
- B) Sum and Product
- C) Average and Sum
- D) First and Last element

Answer: A) Maximum and Minimum

Q76. Count frequency of each element in an array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N
        COUNT = 0
```

```
    FOR J = 1 TO N
        IF ARR[I] = ARR[J]
THEN
            COUNT = COUNT + 1
        ENDIF
    END FOR
    OUTPUT ARR[I], COUNT
END FOR
END
```

Options:

- A) Count frequency ☒
- B) Sum elements
- C) Product elements
- D) Reverse array

Answer: A) Count frequency

Q77. Sort an array using bubble sort.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
```

```
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N-1
        FOR J = 1 TO N-I
            IF ARR[J] > ARR[J+1]
THEN
                TEMP = ARR[J]
                ARR[J] = ARR[J+1]
                ARR[J+1] = TEMP
            ENDIF
        END FOR
    END FOR
    FOR I = 1 TO N
        OUTPUT ARR[I]
    END FOR
END
```

Options:

- A) Bubble sort ☒
- B) Selection sort
- C) Insertion sort
- D) Quick sort

Answer: A) Bubble sort

Q78. Find all prime factors of a number.

Pseudocode:

```
START
  INPUT N
  I = 2
  WHILE N > 1
    IF N MOD I = 0 THEN
      OUTPUT I
      N = N / I
    ELSE
      I = I + 1
    ENDIF
  END WHILE
END
```

Options:

- A) Prime factors ☒
- B) Even numbers
- C) Odd numbers
- D) Fibonacci numbers

Answer: A) Prime factors

Q79. Check if two strings are anagrams.

Pseudocode:

```
START
  INPUT STR1, STR2
  SORT STR1
  SORT STR2
  IF STR1 = STR2 THEN
    OUTPUT "Anagram"
  ELSE
    OUTPUT "Not Anagram"
  ENDIF
END
```

Options:

- A) Anagram or Not ☒
- B) Palindrome or Not
- C) Prime or Not
- D) Even or Odd

Answer: A) Anagram or Not

Q80. Find sum of diagonal elements in a square matrix.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    FOR J = 1 TO N
      INPUT MATRIX[I][J]
      IF I = J THEN
        SUM = SUM +
MATRIX[I][J]
      ENDIF
    END FOR
  END FOR
  OUTPUT SUM
END
```

Options:

- A) Sum of diagonal ☒
- B) Sum of all elements
- C) Product of diagonal

D) Maximum element

Answer: A) Sum of diagonal

Q81. Check if a matrix is symmetric.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    FOR J = 1 TO N
      INPUT MATRIX[I][J]
    END FOR
  END FOR
  FLAG = 0
  FOR I = 1 TO N
    FOR J = 1 TO N
      IF MATRIX[I][J] !=
MATRIX[J][I] THEN
        FLAG = 1
        BREAK
      ENDIF
    END FOR
  END FOR
```

```
END FOR
IF FLAG = 0 THEN
    OUTPUT "Symmetric"
ELSE
    OUTPUT "Not Symmetric"
ENDIF
END
```

Options:

- A) Symmetric or Not ☒
- B) Diagonal sum
- C) Transpose
- D) Maximum element

Answer: A) Symmetric or Not

Q82. Find transpose of a matrix.

Pseudocode:

```
START
    INPUT ROWS, COLS
    FOR I = 1 TO ROWS
        FOR J = 1 TO COLS
            INPUT MATRIX[I][J]
```

```
        END FOR
    END FOR
    FOR I = 1 TO COLS
        FOR J = 1 TO ROWS
            OUTPUT MATRIX[J][I]
        END FOR
    END FOR
END
```

Options:

- A) Transpose ☒
- B) Sum of elements
- C) Symmetric check
- D) Maximum element

Answer: A) Transpose

Q83. Find sum of elements above main diagonal.

Pseudocode:

```
START
    INPUT N
    SUM = 0
```

```
FOR I = 1 TO N
  FOR J = 1 TO N
    INPUT MATRIX[I][J]
    IF J > I THEN
      SUM = SUM +
MATRIX[I][J]
    ENDIF
  END FOR
END FOR
OUTPUT SUM
END
```

Options:

- A) Sum above diagonal ☒
- B) Sum below diagonal
- C) Sum of main diagonal
- D) Sum of all elements

Answer: A) Sum above diagonal

Q84. Rotate a matrix 90 degrees clockwise.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    FOR J = 1 TO N
      INPUT MATRIX[I][J]
    END FOR
  END FOR
  FOR I = 1 TO N
    FOR J = 1 TO N
      ROTATED[J][N-I+1] =
MATRIX[I][J]
    END FOR
  END FOR
  FOR I = 1 TO N
    FOR J = 1 TO N
      OUTPUT ROTATED[I][J]
    END FOR
  END FOR
END
```

Options:

- A) Rotate 90 degrees ☒
- B) Rotate 180 degrees
- C) Transpose

D) Symmetric check

Answer: A) Rotate 90 degrees

Q85. Find majority element in an array (appearing $> N/2$ times).

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N
        COUNT = 0
        FOR J = 1 TO N
            IF ARR[I] = ARR[J]
THEN
                COUNT = COUNT + 1
            ENDIF
        END FOR
        IF COUNT > N/2 THEN
            OUTPUT ARR[I]
```

```
BREAK
ENDIF
END FOR
END
```

Options:

- A) Majority element ☒
- B) Maximum element
- C) Minimum element
- D) Median element

Answer: A) Majority element

Q86. Find maximum difference between two elements ($j > i$).

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  MIN_ELEMENT = ARR[1]
  MAX_DIFF = ARR[2] - ARR[1]
```

```
FOR I = 2 TO N
    IF ARR[I] -
MIN_ELEMENT > MAX_DIFF THEN
        MAX_DIFF = ARR[I] -
MIN_ELEMENT
    ENDIF
    IF ARR[I] < MIN_ELEMENT
THEN
        MIN_ELEMENT = ARR[I]
    ENDIF
END FOR
OUTPUT MAX_DIFF
END
```

Options:

- A) Maximum difference ☒
- B) Minimum difference
- C) Sum difference
- D) Product difference

Answer: A) Maximum difference

Q87. Find first repeating element in an array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N
        FOR J = I+1 TO N
            IF ARR[I] = ARR[J]
THEN
                OUTPUT ARR[I]
                BREAK
            ENDIF
        END FOR
    END FOR
END
```

Options:

- A) First repeating element ☒
- B) First non-repeating
- C) Maximum element

D) Minimum element

Answer: A) First repeating element

Q88. Check if array is sorted.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  FLAG = 0
  FOR I = 1 TO N-1
    IF ARR[I] > ARR[I+1]
THEN
      FLAG = 1
      BREAK
    ENDIF
  END FOR
  IF FLAG = 0 THEN
    OUTPUT "Sorted"
  ELSE
```

```
        OUTPUT "Not Sorted"
    ENDIF
END
```

Options:

- A) Sorted or Not ☒
- B) Reversed or Not
- C) Maximum or Minimum
- D) Sum check

Answer: A) Sorted or Not

Q89. Find length of longest increasing subsequence (LIS) (simple $O(N^2)$ approach).

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
        LIS[I] = 1
    END FOR
```

```
FOR I = 2 TO N
  FOR J = 1 TO I-1
    IF ARR[I] > ARR[J] AND
LIS[I] < LIS[J] + 1 THEN
      LIS[I] = LIS[J] + 1
    ENDIF
  END FOR
END FOR
MAX_LIS = LIS[1]
FOR I = 2 TO N
  IF LIS[I] > MAX_LIS THEN
    MAX_LIS = LIS[I]
  ENDIF
END FOR
OUTPUT MAX_LIS
END
```

Options:

- A) Length of LIS ☒
- B) Sum of array
- C) Maximum element
- D) Minimum element

Answer: A) Length of LIS

Q90. Find missing number in arithmetic progression.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    DIFF = (ARR[N] - ARR[1]) /
N
    FOR I = 1 TO N-1
        IF ARR[I+1] - ARR[I] !=
DIFF THEN
            MISSING = ARR[I] +
DIFF
            OUTPUT MISSING
            BREAK
        ENDIF
    END FOR
END
```

Options:

A) Missing number ☒

B) Maximum element

C) Minimum element

D) Sum of elements

Answer: A) Missing number

Q91. Move all zeros to end of array.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 INPUT ARR[I]

 END FOR

 COUNT = 0

 FOR I = 1 TO N

 IF ARR[I] != 0 THEN

 ARR[COUNT+1] = ARR[I]

 COUNT = COUNT + 1

 ENDIF

 END FOR

```
WHILE COUNT < N
    COUNT = COUNT + 1
    ARR[COUNT] = 0
END WHILE
FOR I = 1 TO N
    OUTPUT ARR[I]
END FOR
END
```

Options:

- A) Move zeros to end ☒
- B) Move zeros to start
- C) Reverse array
- D) Sort array

Answer: A) Move zeros to end

Q92. Find two elements whose sum is closest to zero.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
```

```

        INPUT ARR[I]
    END FOR
    MIN_SUM = INF
    FOR I = 1 TO N-1
        FOR J = I+1 TO N
            IF ABS (ARR[I] + ARR[J])
< MIN_SUM THEN
                MIN_SUM = ABS (ARR[I]
+ ARR[J])
                E1 = ARR[I]
                E2 = ARR[J]
            ENDIF
        END FOR
    END FOR
    OUTPUT E1, E2
END

```

Options:

- A) Pair with sum closest to zero ☒
- B) Pair with maximum sum
- C) Pair with minimum sum
- D) Pair with product closest to zero

Answer: A) Pair with sum closest to zero

Q93. Check if number is perfect number.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N-1
    IF N MOD I = 0 THEN
      SUM = SUM + I
    ENDIF
  END FOR
  IF SUM = N THEN
    OUTPUT "Perfect"
  ELSE
    OUTPUT "Not Perfect"
  ENDIF
END
```

Options:

- A) Perfect or Not ☒
- B) Prime or Not
- C) Armstrong or Not

D) Even or Odd

Answer: A) Perfect or Not

Q94. Generate Pascal's triangle up to N rows.

Pseudocode:

```
START
  INPUT N
  FOR I = 0 TO N-1
    C = 1
    FOR J = 0 TO I
      OUTPUT C
       $C = C * (I - J) / (J + 1)$ 
    END FOR
    OUTPUT NEWLINE
  END FOR
END
```

Options:

A) Pascal's triangle ☒

B) Fibonacci series

C) Prime numbers

D) Even numbers

Answer: A) Pascal's triangle

Q95. Count number of vowels in a string.

Pseudocode:

START

 INPUT STR

 COUNT = 0

 FOR I = 1 TO LENGTH (STR)

 IF STR[I] IN

 ['A', 'E', 'I', 'O', 'U', 'a', 'e',
 'i', 'o', 'u'] THEN

 COUNT = COUNT + 1

 ENDIF

 END FOR

 OUTPUT COUNT

END

Options:

A) Number of vowels ☒

B) Number of consonants

- C) Length of string
- D) Number of words

Answer: A) Number of vowels

Q96. Reverse a string.

Pseudocode:

```
START
  INPUT STR
  REV = ""
  FOR I = LENGTH (STR) DOWNT0
1    REV = REV + STR[I]
  END FOR
  OUTPUT REV
END
```

Options:

- A) Reverse string ☒
- B) Original string
- C) Uppercase string

D) Lowercase string

Answer: A) Reverse string

Q97. Count frequency of characters in string.

Pseudocode:

```
START
  INPUT STR
  FOR I = 1 TO LENGTH (STR)
    COUNT = 0
    FOR J = 1 TO LENGTH (STR)
      IF STR[I] = STR[J]
THEN
          COUNT = COUNT + 1
        ENDIF
      END FOR
      OUTPUT STR[I], COUNT
    END FOR
  END
```

Options:

A) Character frequency ☒

B) Word frequency

C) String length

D) Palindrome check

Answer: A) Character frequency

Q98. Remove duplicate elements from array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    NEW_ARR = []
    FOR I = 1 TO N
        IF ARR[I] NOT IN NEW_ARR
            THEN
                APPEND ARR[I] TO
NEW_ARR
            ENDIF
        END FOR
```

```
    OUTPUT NEW_ARR
END
```

Options:

A) Remove duplicates ☒

B) Reverse array

C) Sort array

D) Sum array

Answer: A) Remove duplicates

Q99. Find longest word in a string.

Pseudocode:

```
START
    INPUT STR
    WORDS = SPLIT(STR)
    MAX_LEN = 0
    FOR EACH WORD IN WORDS
        IF LENGTH(WORD) >
MAX_LEN THEN
            MAX_LEN = LENGTH(WORD)
            LONGEST = WORD
        ENDIF
```

```
END FOR
OUTPUT LONGEST
END
```

Options:

- A) Longest word ☒
- B) Shortest word
- C) Total words
- D) Total characters

Answer: A) Longest word

Q100. Check if a string is palindrome.

Pseudocode:

```
START
  INPUT STR
  REV = ""
  FOR I = LENGTH(STR) DOWNTO
1    REV = REV + STR[I]
  END FOR
  IF STR = REV THEN
    OUTPUT "Palindrome"
```

```
ELSE
    OUTPUT "Not Palindrome"
ENDIF
END
```

Options:

A) Palindrome or Not ☒

B) Uppercase or Not

C) Lowercase or Not

D) Anagram or Not

Answer: A) Palindrome or Not

Q101. Find GCD of two numbers.

Pseudocode:

```
START
    INPUT A, B
    WHILE B != 0
        TEMP = B
        B = A MOD B
        A = TEMP
    END WHILE
```

```
    OUTPUT A
END
```

Options:

A) LCM

B) GCD ☒

C) Sum

D) Product

Answer: B) GCD

Q102. Find LCM of two numbers.

Pseudocode:

```
START
    INPUT A, B
    LCM = (A * B) / GCD(A, B)
    OUTPUT LCM
END
```

Options:

A) GCD

B) LCM ☒

C) Sum

D) Product

Answer: B) LCM

Q103. Check if a number is Armstrong number.

Pseudocode:

START

 INPUT N

 SUM = 0

 TEMP = N

 WHILE TEMP > 0

 DIGIT = TEMP MOD 10

 SUM = SUM + DIGIT³

 TEMP = TEMP / 10

 END WHILE

 IF SUM = N THEN

 OUTPUT "Armstrong"

 ELSE

 OUTPUT "Not Armstrong"

 ENDIF

END

Options:

A) Armstrong or Not ☒

B) Prime or Not

C) Perfect or Not

D) Even or Odd

Answer: A) Armstrong or Not

Q104. Fibonacci series up to N terms.

Pseudocode:

```
START
  INPUT N
  A = 0
  B = 1
  OUTPUT A, B
  FOR I = 3 TO N
    C = A + B
    OUTPUT C
    A = B
    B = C
  END FOR
END
```

Options:

A) Fibonacci series ☒

B) Prime series

C) Even numbers

D) Odd numbers

Answer: A) Fibonacci series

Q105. Find sum of digits of a number.

Pseudocode:

START

 INPUT N

 SUM = 0

 WHILE N > 0

 SUM = SUM + N MOD 10

 N = N / 10

 END WHILE

 OUTPUT SUM

END

Options:

A) Sum of digits ☒

B) Product of digits

C) Reverse number

D) Count digits

Answer: A) Sum of digits

Q106. Reverse a number.

Pseudocode:

START

 INPUT N

 REV = 0

 WHILE N > 0

 REV = REV * 10 + N MOD
10

 N = N / 10

 END WHILE

 OUTPUT REV

END

Options:

A) Reverse number ☒

B) Sum of digits

C) Product of digits

D) Count digits

Answer: A) Reverse number

Q107. Check if number is palindrome.

Pseudocode:

START

 INPUT N

 TEMP = N

 REV = 0

 WHILE TEMP > 0

 REV = REV * 10 + TEMP

MOD 10

 TEMP = TEMP / 10

 END WHILE

 IF N = REV THEN

 OUTPUT "Palindrome"

 ELSE

 OUTPUT "Not Palindrome"

 ENDIF

END

Options:

A) Palindrome or Not ☒

B) Armstrong or Not

C) Prime or Not

D) Even or Odd

Answer: A) Palindrome or Not

Q108. Find factorial of a number.

Pseudocode:

START

 INPUT N

 FACT = 1

 FOR I = 1 TO N

 FACT = FACT * I

 END FOR

 OUTPUT FACT

END

Options:

A) Factorial ☒

B) Sum

C) Product

D) Reverse

Answer: A) Factorial

Q109. Check if number is prime.

Pseudocode:

```
START
  INPUT N
  FLAG = 0
  FOR I = 2 TO N-1
    IF N MOD I = 0 THEN
      FLAG = 1
      BREAK
    ENDIF
  END FOR
  IF FLAG = 0 AND N > 1 THEN
    OUTPUT "Prime"
  ELSE
    OUTPUT "Not Prime"
  ENDIF
END
```

Options:

- A) Prime or Not ☒
- B) Armstrong or Not
- C) Perfect or Not
- D) Even or Odd

Answer: A) Prime or Not

Q110. Find sum of first N natural numbers.

Pseudocode:

```
START
    INPUT N
    SUM = N * (N + 1) / 2
    OUTPUT SUM
END
```

Options:

- A) Sum ☒
- B) Product
- C) Factorial

D) Average

Answer: A) Sum

Q111. Find sum of even numbers in an array.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    INPUT ARR[I]
    IF ARR[I] MOD 2 = 0 THEN
      SUM = SUM + ARR[I]
    ENDIF
  END FOR
  OUTPUT SUM
END
```

Options:

A) Sum of even numbers ☒

B) Sum of odd numbers

C) Maximum element

D) Minimum element

Answer: A) Sum of even numbers

Q112. Find sum of odd numbers in an array.

Pseudocode:

```
START
    INPUT N
    SUM = 0
    FOR I = 1 TO N
        INPUT ARR[I]
        IF ARR[I] MOD 2 != 0
THEN
            SUM = SUM + ARR[I]
        ENDIF
    END FOR
    OUTPUT SUM
END
```

Options:

A) Sum of odd numbers ☒

B) Sum of even numbers

C) Maximum element

D) Minimum element

Answer: A) Sum of odd numbers

Q113. Find second smallest element in an array.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 INPUT ARR[I]

 END FOR

 MIN1 = INF

 MIN2 = INF

 FOR I = 1 TO N

 IF ARR[I] < MIN1 THEN

 MIN2 = MIN1

 MIN1 = ARR[I]

 ELSE IF ARR[I] < MIN2

AND ARR[I] != MIN1 THEN

 MIN2 = ARR[I]

```
        ENDIF
    END FOR
    OUTPUT MIN2
END
```

Options:

- A) Second smallest ☒
- B) Second largest
- C) Smallest
- D) Largest

Answer: A) Second smallest

Q114. Count number of digits in a number.

Pseudocode:

```
START
    INPUT N
    COUNT = 0
    WHILE N > 0
        N = N / 10
        COUNT = COUNT + 1
    END WHILE
```

```
    OUTPUT COUNT
END
```

Options:

A) Number of digits ☒

B) Sum of digits

C) Product of digits

D) Reverse number

Answer: A) Number of digits

Q115. Find sum of all elements in an array.

Pseudocode:

```
START
    INPUT N
    SUM = 0
    FOR I = 1 TO N
        INPUT ARR[I]
        SUM = SUM + ARR[I]
    END FOR
    OUTPUT SUM
END
```

Options:

A) Sum ☒

B) Product

C) Maximum

D) Minimum

Answer: A) Sum

Q116. Find product of all elements in an array.

Pseudocode:

START

 INPUT N

 PRODUCT = 1

 FOR I = 1 TO N

 INPUT ARR[I]

 PRODUCT = PRODUCT *

ARR[I]

 END FOR

 OUTPUT PRODUCT

END

Options:

A) Product ☒

B) Sum

C) Maximum

D) Minimum

Answer: A) Product

Q117. Find maximum element in an array.

Pseudocode:

START

 INPUT N

 FOR I = 1 TO N

 INPUT ARR[I]

 END FOR

 MAX = ARR[1]

 FOR I = 2 TO N

 IF ARR[I] > MAX THEN

 MAX = ARR[I]

 ENDIF

```
END FOR
OUTPUT MAX
END
```

Options:

- A) Maximum ☒
- B) Minimum
- C) Sum
- D) Product

Answer: A) Maximum

Q118. Find minimum element in an array.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  MIN = ARR[1]
  FOR I = 2 TO N
    IF ARR[I] < MIN THEN
```

```
        MIN = ARR[I]
    ENDIF
END FOR
OUTPUT MIN
END
```

Options:

- A) Minimum ☒
- B) Maximum
- C) Sum
- D) Product

Answer: A) Minimum

Q119. Find second largest element in an array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    MAX1 = -INF
```

```
MAX2 = -INF
FOR I = 1 TO N
    IF ARR[I] > MAX1 THEN
        MAX2 = MAX1
        MAX1 = ARR[I]
    ELSE IF ARR[I] > MAX2
AND ARR[I] != MAX1 THEN
        MAX2 = ARR[I]
    ENDIF
END FOR
OUTPUT MAX2
END
```

Options:

- A) Second largest ☒
- B) Largest
- C) Smallest
- D) Sum

Answer: A) Second largest

Q120. Find all prime numbers up to N.

Pseudocode:

```
START
  INPUT N
  FOR I = 2 TO N
    FLAG = 0
    FOR J = 2 TO I-1
      IF I MOD J = 0 THEN
        FLAG = 1
        BREAK
      ENDIF
    END FOR
    IF FLAG = 0 THEN
      OUTPUT I
    ENDIF
  END FOR
END
```

Options:

- A) Prime numbers ☒
- B) Odd numbers
- C) Even numbers
- D) Fibonacci numbers

Answer: A) Prime numbers

Q121. Find sum of prime numbers in an array.

Pseudocode:

```
START
    INPUT N
    SUM = 0
    FOR I = 1 TO N
        INPUT ARR[I]
        FLAG = 0
        FOR J = 2 TO ARR[I]-1
            IF ARR[I] MOD J = 0
THEN
                FLAG = 1
                BREAK
            ENDIF
        END FOR
        IF FLAG = 0 AND ARR[I] >
1 THEN
            SUM = SUM + ARR[I]
        ENDIF
    END FOR
    OUTPUT SUM
```

END

Options:

A) Sum of primes ☒

B) Sum of evens

C) Sum of odds

D) Maximum element

Answer: A) Sum of primes

Q122. Count even and odd numbers in an array.

Pseudocode:

START

INPUT N

EVEN = 0

ODD = 0

FOR I = 1 TO N

INPUT ARR[I]

IF ARR[I] MOD 2 = 0 THEN

EVEN = EVEN + 1

ELSE

ODD = ODD + 1

```
        ENDIF
    END FOR
    OUTPUT EVEN, ODD
END
```

Options:

- A) Count even and odd ☒
- B) Count primes
- C) Count positives
- D) Count negatives

Answer: A) Count even and odd

Q123. Find sum of diagonal elements in a matrix.

Pseudocode:

```
START
    INPUT ROWS, COLS
    SUM = 0
    FOR I = 1 TO ROWS
        FOR J = 1 TO COLS
            INPUT MATRIX[I][J]
            IF I = J THEN
```

```
        SUM = SUM +  
MATRIX[I][J]  
    ENDIF  
END FOR  
END FOR  
OUTPUT SUM  
END
```

Options:

- A) Sum of diagonal ☒
- B) Sum of all elements
- C) Maximum element
- D) Minimum element

Answer: A) Sum of diagonal

Q124. Find transpose of a matrix.

Pseudocode:

```
START  
    INPUT ROWS, COLS  
    FOR I = 1 TO ROWS  
        FOR J = 1 TO COLS  
            INPUT MATRIX[I][J]
```

```
        END FOR
    END FOR
    FOR I = 1 TO COLS
        FOR J = 1 TO ROWS
            OUTPUT MATRIX[J][I]
        END FOR
    END FOR
END
```

Options:

- A) Transpose ☒
- B) Sum
- C) Symmetric check
- D) Maximum element

Answer: A) Transpose

Q125. Check if matrix is symmetric.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        FOR J = 1 TO N
```

```
        INPUT MATRIX[I][J]
    END FOR
END FOR
FLAG = 0
FOR I = 1 TO N
    FOR J = 1 TO N
        IF MATRIX[I][J] !=
MATRIX[J][I] THEN
            FLAG = 1
            BREAK
        ENDIF
    END FOR
END FOR
IF FLAG = 0 THEN
    OUTPUT "Symmetric"
ELSE
    OUTPUT "Not Symmetric"
ENDIF
END
```

Options:

- A) Symmetric or Not ☒
- B) Transpose
- C) Sum

D) Maximum element

Answer: A) Symmetric or Not

Q126. Find sum of elements above main diagonal.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    FOR J = 1 TO N
      INPUT MATRIX[I][J]
      IF J > I THEN
        SUM = SUM +
MATRIX[I][J]
      ENDIF
    END FOR
  END FOR
  OUTPUT SUM
END
```

Options:

- A) Sum above diagonal ☒
- B) Sum below diagonal
- C) Sum of main diagonal
- D) Sum of all elements

Answer: A) Sum above diagonal

Q127. Rotate a matrix 90 degrees clockwise.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    FOR J = 1 TO N
      INPUT MATRIX[I][J]
    END FOR
  END FOR
  FOR I = 1 TO N
    FOR J = 1 TO N
      ROTATED[J][N-I+1] =
MATRIX[I][J]
```

```
        END FOR
    END FOR
    FOR I = 1 TO N
        FOR J = 1 TO N
            OUTPUT ROTATED[I] [J]
        END FOR
    END FOR
END
```

Options:

- A) Rotate 90 degrees ☒
- B) Rotate 180 degrees
- C) Transpose
- D) Symmetric check

Answer: A) Rotate 90 degrees

Q128. Find largest row sum in a matrix.

Pseudocode:

```
START
    INPUT ROWS, COLS
    MAX_SUM = -INF
    FOR I = 1 TO ROWS
```

```
SUM = 0
FOR J = 1 TO COLS
    INPUT MATRIX[I][J]
    SUM = SUM +
MATRIX[I][J]
END FOR
IF SUM > MAX_SUM THEN
    MAX_SUM = SUM
ENDIF
END FOR
OUTPUT MAX_SUM
END
```

Options:

- A) Largest row sum ☒
- B) Largest column sum
- C) Sum of diagonal
- D) Maximum element

Answer: A) Largest row sum

Q129. Merge two sorted arrays.

Pseudocode:

```
START
  INPUT N, M
  FOR I = 1 TO N
    INPUT ARR1[I]
  END FOR
  FOR I = 1 TO M
    INPUT ARR2[I]
  END FOR
  i = 1, j = 1
  WHILE i <= N AND j <= M
    IF ARR1[i] < ARR2[j]
THEN
      OUTPUT ARR1[i]
      i = i + 1
    ELSE
      OUTPUT ARR2[j]
      j = j + 1
    ENDIF
  END WHILE
  WHILE i <= N
    OUTPUT ARR1[i]
    i = i + 1
  END WHILE
```

```
WHILE j <= M
    OUTPUT ARR2[j]
    j = j + 1
END WHILE
END
```

Options:

A) Merge arrays ☒

B) Intersection

C) Difference

D) Reverse arrays

Answer: A) Merge arrays

Q130. Find majority element in an array ($> N/2$ times).

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N
```

```
COUNT = 0
FOR J = 1 TO N
    IF ARR[I] = ARR[J]
THEN
        COUNT = COUNT + 1
    ENDIF
END FOR
IF COUNT > N/2 THEN
    OUTPUT ARR[I]
    BREAK
ENDIF
END FOR
END
```

Options:

- A) Majority element ☒
- B) Maximum element
- C) Minimum element
- D) Median element

Answer: A) Majority element

Q131. Find second largest element in an array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    MAX1 = -INF
    MAX2 = -INF
    FOR I = 1 TO N
        IF ARR[I] > MAX1 THEN
            MAX2 = MAX1
            MAX1 = ARR[I]
        ELSE IF ARR[I] > MAX2
AND ARR[I] != MAX1 THEN
            MAX2 = ARR[I]
        ENDIF
    END FOR
    OUTPUT MAX2
END
```

Options:

A) Second largest ☒

B) Second smallest

C) Largest

D) Smallest

Answer: A) Second largest

Q132. Count number of words in a string.

Pseudocode:

START

 INPUT STR

 WORDS = SPLIT (STR)

 COUNT = LENGTH (WORDS)

 OUTPUT COUNT

END

Options:

A) Number of words ☒

B) Number of characters

C) Number of vowels

D) Number of consonants

Answer: A) Number of words

Q133. Count vowels and consonants in a string.

Pseudocode:

```
START
  INPUT STR
  VOWELS = 0
  CONSONANTS = 0
  FOR I = 1 TO LENGTH (STR)
    CHAR = STR[I]
    IF CHAR IN
['A', 'E', 'I', 'O', 'U', 'a', 'e',
'i', 'o', 'u'] THEN
      VOWELS = VOWELS + 1
    ELSE IF CHAR IS ALPHABET
THEN
      CONSONANTS =
CONSONANTS + 1
    ENDIF
```

```
END FOR
  OUTPUT VOWELS, CONSONANTS
END
```

Options:

- A) Count vowels & consonants ☒
- B) Count digits
- C) Count spaces
- D) Count special characters

Answer: A) Count vowels & consonants

Q134. Reverse words in a string.

Pseudocode:

```
START
  INPUT STR
  WORDS = SPLIT(STR)
  REV_STR = ""
  FOR I = LENGTH(WORDS)
DOWNTO 1
    REV_STR = REV_STR +
WORDS[I] + " "
```

```
    END FOR
    OUTPUT REV_STR
END
```

Options:

- A) Reverse words ☒
- B) Reverse characters
- C) Count words
- D) Count vowels

Answer: A) Reverse words

Q135. Check if string is palindrome.

Pseudocode:

```
START
    INPUT STR
    REV = ""
    FOR I = LENGTH(STR) DOWNT0
1      REV = REV + STR[I]
    END FOR
    IF STR = REV THEN
```

```
        OUTPUT "Palindrome"
ELSE
        OUTPUT "Not Palindrome"
ENDIF
END
```

Options:

- A) Palindrome or Not ☒
- B) Anagram or Not
- C) Uppercase check
- D) Lowercase check

Answer: A) Palindrome or Not

Q136. Find first and last occurrence of an element in array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
```

```
INPUT X
FIRST = -1
LAST = -1
FOR I = 1 TO N
    IF ARR[I] = X THEN
        IF FIRST = -1 THEN
            FIRST = I
        ENDIF
        LAST = I
    ENDIF
END FOR
OUTPUT FIRST, LAST
END
```

Options:

- A) First and last occurrence ☒
- B) Only first occurrence
- C) Only last occurrence
- D) Count occurrence

Answer: A) First and last occurrence

Q137. Find all prime numbers in a range 1 to N.

Pseudocode:

```
START
  INPUT N
  FOR I = 2 TO N
    FLAG = 0
    FOR J = 2 TO SQRT(I)
      IF I MOD J = 0 THEN
        FLAG = 1
        BREAK
      ENDIF
    END FOR
    IF FLAG = 0 THEN
      OUTPUT I
    ENDIF
  END FOR
END
```

Options:

- A) Prime numbers ☒
- B) Even numbers
- C) Odd numbers

D) Fibonacci numbers

Answer: A) Prime numbers

Q138. Sort an array using selection sort.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  FOR I = 1 TO N-1
    MIN_IDX = I
    FOR J = I+1 TO N
      IF ARR[J] <
ARR[MIN_IDX] THEN
        MIN_IDX = J
      ENDIF
    END FOR
    TEMP = ARR[I]
    ARR[I] = ARR[MIN_IDX]
    ARR[MIN_IDX] = TEMP
```

```
END FOR
FOR I = 1 TO N
    OUTPUT ARR[I]
END FOR
END
```

Options:

A) Selection sort ☒

B) Bubble sort

C) Insertion sort

D) Quick sort

Answer: A) Selection sort

Q139. Sort an array using insertion sort.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 2 TO N
```

```
        KEY = ARR[I]
        J = I - 1
        WHILE J > 0 AND ARR[J] >
KEY
            ARR[J+1] = ARR[J]
            J = J - 1
        END WHILE
        ARR[J+1] = KEY
    END FOR
    FOR I = 1 TO N
        OUTPUT ARR[I]
    END FOR
END
```

Options:

- A) Insertion sort ☒
- B) Selection sort
- C) Bubble sort
- D) Quick sort

Answer: A) Insertion sort

Q140. Binary search for an element in sorted array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    INPUT X
    LOW = 1
    HIGH = N
    FOUND = FALSE
    WHILE LOW <= HIGH
        MID = (LOW + HIGH) / 2
        IF ARR[MID] = X THEN
            OUTPUT "Found at
position", MID
            FOUND = TRUE
            BREAK
        ELSE IF ARR[MID] < X
THEN
            LOW = MID + 1
```

```
        ELSE
            HIGH = MID - 1
        ENDIF
    END WHILE
    IF NOT FOUND THEN
        OUTPUT "Not Found"
    ENDIF
END
```

Options:

- A) Binary search ☒
- B) Linear search
- C) Interpolation search
- D) Jump search

Answer: A) Binary search

Q141. Find all factors of a number.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
```

```
        IF N MOD I = 0 THEN
            OUTPUT I
        ENDIF
    END FOR
END
```

Options:

- A) All factors ☒
- B) Prime factors
- C) Even numbers
- D) Odd numbers

Answer: A) All factors

Q142. Find sum of squares of first N natural numbers.

Pseudocode:

```
START
    INPUT N
    SUM = N * (N+1) * (2*N+1) / 6
    OUTPUT SUM
END
```

Options:

A) Sum of squares ☒

B) Sum of cubes

C) Sum

D) Product

Answer: A) Sum of squares

Q143. Generate N Fibonacci numbers using recursion.

Pseudocode:

```
FUNCTION FIB(N)
    IF N = 0 THEN RETURN 0
    ELSE IF N = 1 THEN RETURN
1
    ELSE RETURN FIB(N-1) +
FIB(N-2)
END FUNCTION
```

START

INPUT N

```
FOR I = 0 TO N-1
    OUTPUT FIB(I)
END FOR
END
```

Options:

- A) Fibonacci ☒
- B) Prime
- C) Even
- D) Odd

Answer: A) Fibonacci

Q144. Find sum of digits using recursion.

Pseudocode:

```
FUNCTION SUM_DIGITS(N)
    IF N = 0 THEN RETURN 0
    ELSE RETURN N MOD 10 +
SUM_DIGITS(N/10)
END FUNCTION
```

START

```
INPUT N
OUTPUT SUM_DIGITS (N)
END
```

Options:

- A) Sum of digits ☒
- B) Product of digits
- C) Reverse number
- D) Count digits

Answer: A) Sum of digits

Q145. Sort array using insertion sort.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  FOR I = 2 TO N
    KEY = ARR[I]
```

```
        J = I - 1
        WHILE J > 0 AND ARR[J] >
KEY
            ARR[J+1] = ARR[J]
            J = J - 1
        END WHILE
        ARR[J+1] = KEY
    END FOR
    FOR I = 1 TO N
        OUTPUT ARR[I]
    END FOR
END
```

Options:

- A) Insertion sort ☒
- B) Bubble sort
- C) Selection sort
- D) Quick sort

Answer: A) Insertion sort

Q146. Find median of array.

Pseudocode:

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  SORT ARR
  IF N MOD 2 = 0 THEN
    MEDIAN = (ARR[N/2] +
ARR[N/2 + 1]) / 2
  ELSE
    MEDIAN = ARR[(N+1) / 2]
  ENDIF
  OUTPUT MEDIAN
END
```

Options:

A) Median ☒

B) Mean

C) Mode

D) Range

Answer: A) Median

Q147. Find mode of array (most frequent element).

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    MAX_COUNT = 0
    MODE = ARR[1]
    FOR I = 1 TO N
        COUNT = 0
        FOR J = 1 TO N
            IF ARR[I] = ARR[J]
THEN
                COUNT = COUNT + 1
            ENDIF
        END FOR
        IF COUNT > MAX_COUNT
THEN
            MAX_COUNT = COUNT
            MODE = ARR[I]
```

```
        ENDIF
    END FOR
    OUTPUT MODE
END
```

Options:

A) Mode ☒

B) Median

C) Mean

D) Range

Answer: A) Mode

Q148. Reverse a string.

Pseudocode:

```
START
    INPUT STR
    REV = ""
    FOR I = LENGTH(STR) DOWNTO
1      REV = REV + STR[I]
    END FOR
    OUTPUT REV
```

END

Options:

A) Reverse string ☒

B) Uppercase string

C) Palindrome check

D) Lowercase string

Answer: A) Reverse string

Q149. Check if string is palindrome.

Pseudocode:

START

 INPUT STR

 REV = ""

 FOR I = LENGTH(STR) DOWNTOW
1

 REV = REV + STR[I]

 END FOR

 IF STR = REV THEN

 OUTPUT "Palindrome"

 ELSE

 OUTPUT "Not Palindrome"

```
    ENDIF  
END
```

Options:

- A) Palindrome or Not ☒
- B) Anagram check
- C) Uppercase
- D) Count vowels

Answer: A) Palindrome or Not

Q150. Count vowels in a string.

Pseudocode:

```
START  
    INPUT STR  
    COUNT = 0  
    FOR I = 1 TO LENGTH(STR)  
        IF STR[I] IN  
['A', 'E', 'I', 'O', 'U', 'a', 'e',  
'i', 'o', 'u'] THEN  
            COUNT = COUNT + 1  
        ENDIF  
    END FOR
```

```
    OUTPUT COUNT
END
```

Options:

A) Count vowels ☒

B) Count consonants

C) Count digits

D) Count words

Answer: A) Count vowels

Q151. Count consonants in a string.

Pseudocode:

```
START
    INPUT STR
    COUNT = 0
    FOR I = 1 TO LENGTH(STR)
        IF STR[I] IN
['A'..'Z', 'a'..'z'] AND
STR[I] NOT IN
['A', 'E', 'I', 'O', 'U', 'a', 'e',
'i', 'o', 'u'] THEN
            COUNT = COUNT + 1
```

```
        ENDIF
    END FOR
    OUTPUT COUNT
END
```

Options:

- A) Count consonants ☒
- B) Count vowels
- C) Count digits
- D) Count words

Answer: A) Count consonants

Q152. Convert string to uppercase.

Pseudocode:

```
START
    INPUT STR
    FOR I = 1 TO LENGTH(STR)
        IF STR[I] >= 'a' AND
STR[I] <= 'z' THEN
            STR[I] = STR[I] - 32
        ENDIF
    END FOR
```

```
    OUTPUT STR
END
```

Options:

A) Uppercase ☒

B) Lowercase

C) Reverse string

D) Count vowels

Answer: A) Uppercase

Q153. Convert string to lowercase.

Pseudocode:

```
START
    INPUT STR
    FOR I = 1 TO LENGTH(STR)
        IF STR[I] >= 'A' AND
STR[I] <= 'Z' THEN
            STR[I] = STR[I] + 32
        ENDIF
    END FOR
    OUTPUT STR
END
```

Options:

A) Lowercase ☒

B) Uppercase

C) Reverse string

D) Count vowels

Answer: A) Lowercase

Q154. Find factorial using recursion.

Pseudocode:

```
FUNCTION FACTORIAL (N)
    IF N = 0 OR N = 1 THEN
        RETURN 1
    ELSE
        RETURN N * FACTORIAL (N-1)
    ENDIF
END FUNCTION
```

```
START
    INPUT N
    OUTPUT FACTORIAL (N)
END
```

Options:

A) Factorial ☒

B) Sum

C) Fibonacci

D) Prime check

Answer: A) Factorial

Q155. Find nth Fibonacci number using recursion.

Pseudocode:

```
FUNCTION FIB (N)
    IF N = 0 THEN RETURN 0
    IF N = 1 THEN RETURN 1
    RETURN FIB (N-1) + FIB (N-2)
END FUNCTION
```

```
START
    INPUT N
    OUTPUT FIB (N)
END
```

Options:

A) Fibonacci ☒

B) Factorial

C) Sum

D) Prime check

Answer: A) Fibonacci

Q156. Solve Tower of Hanoi problem.

Pseudocode:

```
FUNCTION TOH (N, FROM, TO,
AUX)
    IF N = 1 THEN
        OUTPUT "Move disk 1 from
" + FROM + " to " + TO
        RETURN
    ENDIF
    TOH (N-1, FROM, AUX, TO)
    OUTPUT "Move disk " + N +
" from " + FROM + " to " +
TO
    TOH (N-1, AUX, TO, FROM)
```

```
END FUNCTION
```

```
START
```

```
    INPUT N
```

```
    TOH (N, 'A', 'C', 'B')
```

```
END
```

Options:

A) Tower of Hanoi ☒

B) Fibonacci

C) Factorial

D) Prime check

Answer: A) Tower of Hanoi

Q157. Sum of digits using recursion.

Pseudocode:

```
FUNCTION SUM_DIGITS (N)
```

```
    IF N = 0 THEN RETURN 0
```

```
    RETURN (N MOD 10) +
```

```
    SUM_DIGITS (N / 10)
```

```
END FUNCTION
```

```
START
    INPUT N
    OUTPUT SUM_DIGITS (N)
END
```

Options:

A) Sum of digits ☒

B) Factorial

C) Fibonacci

D) Product of digits

Answer: A) Sum of digits

Q158. Reverse a number using recursion.

Pseudocode:

```
FUNCTION REVERSE (N, REV)
    IF N = 0 THEN RETURN REV
    RETURN REVERSE (N/10,
REV*10 + N MOD 10)
END FUNCTION
```

START

```
    INPUT N
    OUTPUT REVERSE (N, 0)
END
```

Options:

- A) Reverse number ☒
- B) Sum of digits
- C) Factorial
- D) Fibonacci

Answer: A) Reverse number

Q159. Count digits using recursion.

Pseudocode:

```
FUNCTION COUNT_DIGITS (N)
    IF N = 0 THEN RETURN 0
    RETURN 1 +
COUNT_DIGITS (N/10)
END FUNCTION
```

```
START
    INPUT N
    OUTPUT COUNT_DIGITS (N)
```

END

Options:

A) Count digits ☒

B) Sum of digits

C) Factorial

D) Reverse number

Answer: A) Count digits

Q160. Check palindrome number using recursion.

Pseudocode:

```
FUNCTION IS_PAL(N, REV)
    IF N = 0 THEN RETURN REV
    RETURN IS_PAL(N/10, REV*10
+ N MOD 10)
END FUNCTION
```

START

INPUT N

IF N = IS_PAL(N, 0) THEN

OUTPUT "Palindrome"

```
ELSE
    OUTPUT "Not Palindrome"
ENDIF
END
```

Options:

A) Palindrome ☒

B) Armstrong

C) Prime

D) Factorial

Answer: A) Palindrome

Q161. Find all prime numbers in an array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N
```

```
    FLAG = 0
    IF ARR[I] < 2 THEN
        CONTINUE
    ENDIF
    FOR J = 2 TO Sqrt(ARR[I])
        IF ARR[I] MOD J = 0
THEN
            FLAG = 1
            BREAK
        ENDIF
    END FOR
    IF FLAG = 0 THEN
        OUTPUT ARR[I]
    ENDIF
END FOR
END
```

Options:

- A) Prime numbers ☒
- B) Even numbers
- C) Odd numbers
- D) Composite numbers

Answer: A) Prime numbers

Q162. Find all perfect numbers in an array.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    FOR I = 1 TO N
        SUM = 0
        FOR J = 1 TO ARR[I]-1
            IF ARR[I] MOD J = 0
THEN
                SUM = SUM + J
            ENDIF
        END FOR
        IF SUM = ARR[I] THEN
            OUTPUT ARR[I]
        ENDIF
    END FOR
END
```

Options:

A) Perfect numbers ☒

B) Prime numbers

C) Even numbers

D) Odd numbers

Answer: A) Perfect numbers

Q163. Find largest element in each row of a matrix.

Pseudocode:

START

 INPUT ROWS, COLS

 FOR I = 1 TO ROWS

 FOR J = 1 TO COLS

 INPUT MATRIX[I][J]

 END FOR

 END FOR

 FOR I = 1 TO ROWS

 MAX = MATRIX[I][1]

 FOR J = 2 TO COLS

```
        IF MATRIX[I][J] > MAX
THEN
        MAX = MATRIX[I][J]
        ENDIF
    END FOR
    OUTPUT MAX
END FOR
END
```

Options:

- A) Largest element of each row ☒
- B) Smallest element
- C) Row sum
- D) Column sum

Answer: A) Largest element of each row

Q164. Find smallest element in each column of a matrix.

Pseudocode:

```
START
    INPUT ROWS, COLS
    FOR I = 1 TO ROWS
```

```
    FOR J = 1 TO COLS
        INPUT MATRIX[I][J]
    END FOR
END FOR
FOR J = 1 TO COLS
    MIN = MATRIX[1][J]
    FOR I = 2 TO ROWS
        IF MATRIX[I][J] < MIN
THEN
            MIN = MATRIX[I][J]
        ENDIF
    END FOR
    OUTPUT MIN
END FOR
END
```

Options:

- A) Smallest element of each column ☒
- B) Largest element
- C) Row sum
- D) Column sum

Answer: A) Smallest element of each column

Q165. Rotate matrix 180 degrees.

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
        FOR J = 1 TO N
            INPUT MATRIX[I][J]
        END FOR
    END FOR
    FOR I = 1 TO N
        FOR J = 1 TO N
            ROTATED[N-I+1][N-J+1]
= MATRIX[I][J]
        END FOR
    END FOR
    FOR I = 1 TO N
        FOR J = 1 TO N
            OUTPUT ROTATED[I][J]
        END FOR
    END FOR
END
```

Options:

A) Rotate 180 degrees ☒

B) Rotate 90 degrees

C) Transpose

D) Symmetric check

Answer: A) Rotate 180 degrees

Q166. Find sum of upper triangle elements.

Pseudocode:

START

 INPUT N

 SUM = 0

 FOR I = 1 TO N

 FOR J = 1 TO N

 INPUT MATRIX[I][J]

 IF J > I THEN

 SUM = SUM +

MATRIX[I][J]

 ENDIF

 END FOR

```
END FOR
OUTPUT SUM
END
```

Options:

- A) Sum of upper triangle ☒
- B) Sum of lower triangle
- C) Sum of main diagonal
- D) Sum of all elements

Answer: A) Sum of upper triangle

Q167. Find sum of lower triangle elements.

Pseudocode:

```
START
  INPUT N
  SUM = 0
  FOR I = 1 TO N
    FOR J = 1 TO N
      INPUT MATRIX[I][J]
      IF I > J THEN
```

```
        SUM = SUM +  
MATRIX[I][J]  
    ENDIF  
END FOR  
END FOR  
OUTPUT SUM  
END
```

Options:

- A) Sum of lower triangle ☒
- B) Sum of upper triangle
- C) Sum of diagonal
- D) Sum of all elements

Answer: A) Sum of lower triangle

Q168. Find transpose of non-square matrix.

Pseudocode:

```
START  
    INPUT ROWS, COLS  
    FOR I = 1 TO ROWS  
        FOR J = 1 TO COLS
```

```
        INPUT MATRIX[I] [J]
    END FOR
END FOR
FOR I = 1 TO COLS
    FOR J = 1 TO ROWS
        OUTPUT MATRIX[J] [I]
    END FOR
END FOR
END
```

Options:

- A) Transpose ☒
- B) Sum of elements
- C) Symmetric check
- D) Rotate matrix

Answer: A) Transpose

Q169. Search element in matrix.

Pseudocode:

```
START
    INPUT ROWS, COLS, X
    FOUND = 0
```

```
FOR I = 1 TO ROWS
  FOR J = 1 TO COLS
    INPUT MATRIX[I][J]
    IF MATRIX[I][J] = X
THEN
      OUTPUT "Found at", I,
J
      FOUND = 1
    ENDIF
  END FOR
END FOR
IF FOUND = 0 THEN
  OUTPUT "Not Found"
ENDIF
END
```

Options:

- A) Search in matrix ☒
- B) Sum of elements
- C) Max element
- D) Min element

Answer: A) Search in matrix

Q170. Spiral order of a matrix.

Pseudocode:

```
START
  INPUT ROWS, COLS
  FOR I = 1 TO ROWS
    FOR J = 1 TO COLS
      INPUT MATRIX[I][J]
    END FOR
  END FOR
  TOP = 1, BOTTOM = ROWS,
LEFT = 1, RIGHT = COLS
  WHILE TOP <= BOTTOM AND
LEFT <= RIGHT
    FOR J = LEFT TO RIGHT
      OUTPUT MATRIX[TOP][J]
    END FOR
    TOP = TOP + 1
    FOR I = TOP TO BOTTOM
      OUTPUT MATRIX[I][RIGHT]
    END FOR
    RIGHT = RIGHT - 1
    IF TOP <= BOTTOM THEN
```

```

        FOR J = RIGHT DOWNT0
LEFT
        OUTPUT
MATRIX[BOTTOM] [J]
        END FOR
        BOTTOM = BOTTOM - 1
ENDIF
    IF LEFT <= RIGHT THEN
        FOR I = BOTTOM DOWNT0
TOP
            OUTPUT
MATRIX[I] [LEFT]
            END FOR
            LEFT = LEFT + 1
        ENDIF
    END WHILE
END

```

Options:

- A) Spiral order ☒
- B) Row-wise order
- C) Column-wise order

D) Diagonal order

Answer: A) Spiral order

Q171. Find factorial using recursion.

Pseudocode:

```
FUNCTION FACTORIAL(N)
    IF N = 0 OR N = 1 THEN
        RETURN 1
    ELSE
        RETURN N * FACTORIAL(N-1)
    ENDIF
END FUNCTION
```

```
START
    INPUT N
    OUTPUT FACTORIAL(N)
```

END

Options:

A) Factorial ☒

B) Sum

C) Fibonacci

D) Power

Answer: A) Factorial

Q172. Find Nth Fibonacci using recursion.

Pseudocode:

```
FUNCTION FIB (N)
    IF N = 0 THEN
        RETURN 0
    ELSE IF N = 1 THEN
        RETURN 1
    ELSE
        RETURN FIB (N-1) + FIB (N-
2)
    ENDIF
```

END FUNCTION

START

INPUT N

OUTPUT FIB (N)

END

Options:

A) Fibonacci ☒

B) Factorial

C) Sum of digits

D) Power

Answer: A) Fibonacci

Q173. Reverse a string using recursion.

Pseudocode:

FUNCTION REVERSE (STR)

IF LENGTH (STR) <= 1 THEN

RETURN STR

ELSE

```
        RETURN
    REVERSE (SUBSTRING (STR, 2, LENGTH (STR) - 1)) + STR[1]
    ENDIF
END FUNCTION
```

```
START
    INPUT STR
    OUTPUT REVERSE (STR)
END
```

Options:

- A) Reverse string ☒
- B) Palindrome check
- C) Uppercase conversion
- D) Count vowels

Answer: A) Reverse string

Q174. Check palindrome using recursion.

Pseudocode:

```
FUNCTION IS_PALINDROME (STR)
    IF LENGTH (STR) <= 1 THEN
        RETURN TRUE
    ELSE IF STR[1] !=
STR[LENGTH (STR)] THEN
        RETURN FALSE
    ELSE
        RETURN
IS_PALINDROME (SUBSTRING (STR,
2, LENGTH (STR) - 2))
    ENDIF
END FUNCTION
```

```
START
    INPUT STR
    IF IS_PALINDROME (STR) THEN
        OUTPUT "Palindrome"
    ELSE
        OUTPUT "Not Palindrome"
    ENDIF
END
```

Options:

A) Palindrome or Not ☒

- B) Reverse string
- C) Uppercase string
- D) Count consonants

Answer: A) Palindrome or Not

Q175. Sum of array elements using recursion.

Pseudocode:

```
FUNCTION SUM_ARRAY (ARR, N)
    IF N = 0 THEN
        RETURN 0
    ELSE
        RETURN ARR[N] +
SUM_ARRAY (ARR, N-1)
    ENDIF
END FUNCTION
```

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
```

```
    END FOR
    OUTPUT SUM_ARRAY (ARR, N)
END
```

Options:

A) Sum ☒

B) Product

C) Maximum

D) Minimum

Answer: A) Sum

Q176. Find maximum element using recursion.

Pseudocode:

```
FUNCTION MAX_ARRAY (ARR, N)
    IF N = 1 THEN
        RETURN ARR[1]
    ELSE
        RETURN MAX (ARR[N],
MAX_ARRAY (ARR, N-1) )
    ENDIF
```

```
END FUNCTION
```

```
START
```

```
    INPUT N
```

```
    FOR I = 1 TO N
```

```
        INPUT ARR[I]
```

```
    END FOR
```

```
    OUTPUT MAX_ARRAY (ARR, N)
```

```
END
```

Options:

A) Maximum ☒

B) Minimum

C) Sum

D) Product

Answer: A) Maximum

Q177. Find minimum element using recursion.

Pseudocode:

```
FUNCTION MIN_ARRAY (ARR, N)
```

```
IF N = 1 THEN
    RETURN ARR[1]
ELSE
    RETURN MIN(ARR[N],
MIN_ARRAY(ARR, N-1))
ENDIF
END FUNCTION

START
INPUT N
FOR I = 1 TO N
    INPUT ARR[I]
END FOR
OUTPUT MIN_ARRAY(ARR, N)
END
```

Options:

- A) Minimum ☒
- B) Maximum
- C) Sum
- D) Product

Answer: A) Minimum

Q178. Tower of Hanoi problem.

Pseudocode:

```
FUNCTION TOH (N, FROM, TO,
AUX)
    IF N = 1 THEN
        OUTPUT "Move disk 1 from
", FROM, " to ", TO
        RETURN
    ENDIF
    TOH (N-1, FROM, AUX, TO)
    OUTPUT "Move disk ", N, "
from ", FROM, " to ", TO
    TOH (N-1, AUX, TO, FROM)
END FUNCTION
```

```
START
    INPUT N
    TOH (N, "A", "C", "B")
END
```

Options:

A) Tower of Hanoi ☒

B) Fibonacci

C) Factorial

D) Sum of array

Answer: A) Tower of Hanoi

Q179. Count digits of number using recursion.

Pseudocode:

```
FUNCTION COUNT_DIGITS (N)
    IF N = 0 THEN
        RETURN 0
    ELSE
        RETURN 1 +
COUNT_DIGITS (N/10)
    ENDIF
END FUNCTION
```

START

```
INPUT N
OUTPUT COUNT_DIGITS (N)
END
```

Options:

- A) Count digits ☒
- B) Sum digits
- C) Reverse number
- D) Product digits

Answer: A) Count digits

Q180. Sum of digits using recursion.

Pseudocode:

```
FUNCTION SUM_DIGITS (N)
  IF N = 0 THEN
    RETURN 0
  ELSE
    RETURN (N MOD 10) +
SUM_DIGITS (N/10)
  ENDIF
END FUNCTION
```

```
START
  INPUT N
  OUTPUT SUM_DIGITS (N)
END
```

Options:

- A) Sum digits ☒
- B) Product digits
- C) Count digits
- D) Reverse number

Answer: A) Sum digits

Q181. Power of a number using recursion.

Pseudocode:

```
FUNCTION POWER (A, B)
  IF B = 0 THEN
    RETURN 1
  ELSE
    RETURN A * POWER (A, B-1)
```

```
        ENDIF
    END FUNCTION

START
    INPUT A, B
    OUTPUT POWER(A, B)
END
```

Options:

A) Power ☒

B) Factorial

C) Sum

D) Product

Answer: A) Power

Q182. Count number of zeros in number using recursion.

Pseudocode:

```
FUNCTION COUNT_ZERO(N)
    IF N = 0 THEN
        RETURN 0
```

```
    ELSE IF N MOD 10 = 0 THEN
        RETURN 1 +
COUNT_ZERO (N/10)
    ELSE
        RETURN COUNT_ZERO (N/10)
    ENDIF
END FUNCTION
```

```
START
    INPUT N
    OUTPUT COUNT_ZERO (N)
END
```

Options:

- A) Count zeros ☒
- B) Count digits
- C) Sum digits
- D) Reverse number

Answer: A) Count zeros

Q183. Find GCD using recursion.

Pseudocode:

```
FUNCTION GCD (A, B)
    IF B = 0 THEN
        RETURN A
    ELSE
        RETURN GCD (B, A MOD B)
    ENDIF
END FUNCTION
```

```
START
    INPUT A, B
    OUTPUT GCD (A, B)
END
```

Options:

- A) GCD ☒
- B) LCM
- C) Sum
- D) Product

Answer: A) GCD

Q184. Find LCM using recursion.

Pseudocode:

```
FUNCTION LCM (A, B)
    RETURN (A * B) / GCD (A, B)
END FUNCTION
```

```
START
    INPUT A, B
    OUTPUT LCM (A, B)
END
```

Options:

A) LCM ☒

B) GCD

C) Sum

D) Product

Answer: A) LCM

Q185. Reverse a linked list using recursion (conceptual).

Pseudocode:

```
FUNCTION  
REVERSE_LINKEDLIST (NODE)  
    IF NODE = NULL OR  
NODE.NEXT = NULL THEN  
        RETURN NODE  
    ENDIF  
    NEW_HEAD =  
REVERSE_LINKEDLIST (NODE.NEXT)  
    NODE.NEXT.NEXT = NODE  
    NODE.NEXT = NULL  
    RETURN NEW_HEAD  
END FUNCTION
```

Options:

- A) Reverse linked list ☒
- B) Reverse array
- C) Sum linked list
- D) Count linked list

Answer: A) Reverse linked list

Q186. Find all subsets of a set using recursion.

Pseudocode:

```
FUNCTION SUBSETS (SET, INDEX,
CURRENT)
    IF INDEX = LENGTH (SET)
THEN
        OUTPUT CURRENT
        RETURN
    ENDIF
    SUBSETS (SET, INDEX+1,
CURRENT)
    SUBSETS (SET, INDEX+1,
CURRENT + SET [INDEX])
END FUNCTION
```

```
START
    INPUT SET
    SUBSETS (SET, 0, "")
END
```

Options:

A) All subsets ☒

- B) Permutations
- C) Combinations of 2
- D) Sum of set

Answer: A) All subsets

Q187. Find all permutations of string using recursion.

Pseudocode:

```
FUNCTION PERMUTE (STR, L, R)
    IF L = R THEN
        OUTPUT STR
    ELSE
        FOR I = L TO R
            SWAP STR[L], STR[I]
            PERMUTE (STR, L+1, R)
            SWAP STR[L], STR[I]
        END FOR
    ENDIF
END FUNCTION

START
```

```
INPUT STR
PERMUTE (STR, 0,
LENGTH (STR) - 1)
END
```

Options:

A) Permutations ☒

B) Subsets

C) Reverse

D) Palindrome

Answer: A) Permutations

Q188. Find first missing positive integer.

Pseudocode:

```
START
INPUT N
FOR I = 1 TO N
    INPUT ARR[I]
END FOR
SORT ARR
MISSING = 1
```

```
FOR I = 1 TO N
    IF ARR[I] = MISSING THEN
        MISSING = MISSING + 1
    ENDIF
END FOR
OUTPUT MISSING
END
```

Options:

- A) First missing positive ☒
- B) Maximum element
- C) Minimum element
- D) Sum of array

Answer: A) First missing positive

Q189. Find longest increasing subsequence (conceptual).

Pseudocode:

```
START
    INPUT N
    FOR I = 1 TO N
```

```

        INPUT ARR[I]
    END FOR
    LIS[N] = 1
    FOR I = 2 TO N
        FOR J = 1 TO I-1
            IF ARR[I] > ARR[J] AND
LIS[I] < LIS[J]+1 THEN
                LIS[I] = LIS[J]+1
            ENDIF
        END FOR
    END FOR
    OUTPUT MAX (LIS [1..N] )
END

```

Options:

- A) Longest increasing subsequence ☒
- B) Longest decreasing subsequence
- C) Maximum sum subsequence
- D) Maximum product subsequence

Answer: A) Longest increasing subsequence

Q190. Count number of paths in a matrix (from top-left to bottom-right).

Pseudocode:

```
FUNCTION PATHS (M, N)
    IF M = 1 OR N = 1 THEN
        RETURN 1
    ELSE
        RETURN PATHS (M-1, N) +
PATHS (M, N-1)
    ENDIF
END FUNCTION

START
    INPUT M, N
    OUTPUT PATHS (M, N)
END
```

Options:

- A) Number of paths ☒
- B) Maximum sum path
- C) Minimum sum path

D) Diagonal sum

Answer: A) Number of paths

Q191. Find power of a number using recursion (A^B).

Pseudocode:

```
FUNCTION POWER (A, B)
    IF B = 0 THEN
        RETURN 1
    ELSE
        RETURN A * POWER (A, B-1)
    ENDIF
END FUNCTION
```

```
START
    INPUT A, B
    OUTPUT POWER (A, B)
```

END

Options:

A) Recursive power ☒

B) Iterative power

C) Sum

D) Product

Answer: A) Recursive power

Q192. Reverse a string using recursion.

Pseudocode:

```
FUNCTION REVERSE (STR)
    IF LENGTH (STR) <= 1 THEN
        RETURN STR
    ELSE
        RETURN
        REVERSE (SUBSTRING (STR, 2,
        LENGTH (STR) ) ) + STR[1]
    ENDIF
END FUNCTION

START
```

```
INPUT STR
OUTPUT REVERSE (STR)
END
```

Options:

- A) Recursive reverse ☒
- B) Iterative reverse
- C) Palindrome check
- D) Uppercase conversion

Answer: A) Recursive reverse

Q193. Find sum of digits of a number using recursion.

Pseudocode:

```
FUNCTION SUM_DIGITS (N)
  IF N = 0 THEN
    RETURN 0
  ELSE
    RETURN (N MOD 10) +
SUM_DIGITS (N/10)
  ENDIF
END FUNCTION
```

```
START
  INPUT N
  OUTPUT SUM_DIGITS (N)
END
```

Options:

- A) Recursive sum of digits ☒
- B) Iterative sum
- C) Product of digits
- D) Count digits

Answer: A) Recursive sum of digits

Q194. Check if array is sorted using recursion.

Pseudocode:

```
FUNCTION IS_SORTED (ARR, N)
  IF N = 1 THEN
    RETURN TRUE
  ELSE IF ARR[N] < ARR[N-1]
  THEN
    RETURN FALSE
```

```
        ELSE
            RETURN IS_SORTED (ARR, N-
1)
        ENDIF
END FUNCTION
```

```
START
    INPUT N
    FOR I = 1 TO N
        INPUT ARR[I]
    END FOR
    IF IS_SORTED (ARR, N) THEN
        OUTPUT "Sorted"
    ELSE
        OUTPUT "Not Sorted"
    ENDIF
END
```

Options:

- A) Recursive check sorted ☒
- B) Iterative check
- C) Reverse array

D) Sort array

Answer: A) Recursive check sorted

Q195. Find maximum element in array using recursion.

Pseudocode:

```
FUNCTION MAX_ARRAY (ARR, N)
    IF N = 1 THEN
        RETURN ARR[1]
    ELSE
        MAX_REST = MAX_ARRAY (ARR,
N-1)
        IF ARR[N] > MAX_REST
THEN
            RETURN ARR[N]
        ELSE
            RETURN MAX_REST
        ENDIF
    ENDIF
END FUNCTION
```

```
START
  INPUT N
  FOR I = 1 TO N
    INPUT ARR[I]
  END FOR
  OUTPUT MAX_ARRAY (ARR, N)
END
```

Options:

- A) Recursive maximum ☒
- B) Iterative maximum
- C) Minimum
- D) Sum

Answer: A) Recursive maximum

Q196. Find minimum element in array using recursion.

Pseudocode:

```
FUNCTION MIN_ARRAY (ARR, N)
  IF N = 1 THEN
    RETURN ARR[1]
  ELSE
```

```
        MIN_REST = MIN_ARRAY (ARR,  
N-1)  
        IF ARR[N] < MIN_REST  
THEN  
            RETURN ARR[N]  
        ELSE  
            RETURN MIN_REST  
        ENDIF  
    ENDIF  
END FUNCTION
```

```
START  
    INPUT N  
    FOR I = 1 TO N  
        INPUT ARR[I]  
    END FOR  
    OUTPUT MIN_ARRAY (ARR, N)  
END
```

Options:

- A) Recursive minimum ☒
- B) Iterative minimum
- C) Maximum

D) Sum

Answer: A) Recursive minimum

Q197. Sum of elements in a 2D matrix using recursion.

Pseudocode:

```
FUNCTION SUM_MATRIX (MATRIX,  
ROWS, COLS)  
    IF ROWS = 0 THEN  
        RETURN 0  
    ELSE IF COLS = 0 THEN  
        RETURN SUM_MATRIX (MATRIX,  
ROWS-1, ORIGINAL_COLS)  
    ELSE  
        RETURN MATRIX[ROWS][COLS]  
+ SUM_MATRIX (MATRIX, ROWS,  
COLS-1)  
    ENDIF  
END FUNCTION  
  
START
```

```
INPUT ROWS, COLS
FOR I = 1 TO ROWS
    FOR J = 1 TO COLS
        INPUT MATRIX[I][J]
    END FOR
END FOR
OUTPUT SUM_MATRIX(MATRIX,
ROWS, COLS)
END
```

Options:

- A) Recursive sum of matrix ☒
- B) Iterative sum
- C) Maximum
- D) Minimum

Answer: A) Recursive sum of matrix

Q198. Find transpose of matrix using recursion.

Pseudocode:

```
FUNCTION TRANSPOSE(MATRIX,
ROW, COL, NEW_MATRIX)
```

```
    IF ROW = TOTAL_ROWS THEN
        RETURN
    ELSE IF COL = TOTAL_COLS
THEN
        TRANSPOSE (MATRIX, ROW+1,
0, NEW_MATRIX)
    ELSE
        NEW_MATRIX[COL][ROW] =
MATRIX[ROW][COL]
        TRANSPOSE (MATRIX, ROW,
COL+1, NEW_MATRIX)
    ENDIF
END FUNCTION
```

```
START
    INPUT ROWS, COLS
    FOR I = 1 TO ROWS
        FOR J = 1 TO COLS
            INPUT MATRIX[I][J]
        END FOR
    END FOR
    TRANSPOSE (MATRIX, 0, 0,
NEW_MATRIX)
```

```
FOR I = 0 TO COLS-1
  FOR J = 0 TO ROWS-1
    OUTPUT NEW_MATRIX[I][J]
  END FOR
END FOR
END
```

Options:

- A) Recursive transpose ☒
- B) Iterative transpose
- C) Sum matrix
- D) Rotate matrix

Answer: A) Recursive transpose

Q199. Find all paths in a maze (backtracking).

Pseudocode:

```
FUNCTION FIND_PATH(MAZE, X,  
Y, PATH)  
  IF X = END_X AND Y = END_Y  
  THEN  
    OUTPUT PATH
```

```
        RETURN
    ENDIF
    MARK MAZE[X][Y] AS VISITED
    FOR EACH MOVE IN [UP, DOWN,
LEFT, RIGHT]
        IF MOVE IS VALID THEN
            FIND_PATH(MAZE, NEW_X,
NEW_Y, PATH + MOVE)
        ENDIF
    END FOR
    UNMARK MAZE[X][Y]
END FUNCTION

START
    INPUT MAZE
    CALL FIND_PATH(MAZE,
START_X, START_Y, "")
END
```

Options:

- A) Backtracking paths ☒
- B) DFS only
- C) BFS only

D) Matrix sum

Answer: A) Backtracking paths

Q200. Solve N-Queens problem using recursion/backtracking.

Pseudocode:

```
FUNCTION PLACE_QUEENS (ROW)
    IF ROW > N THEN
        OUTPUT BOARD
        RETURN
    ENDIF
    FOR COL = 1 TO N
        IF SAFE (ROW, COL) THEN
            PLACE QUEEN AT
BOARD[ROW][COL]
            PLACE_QUEENS (ROW+1)
            REMOVE QUEEN FROM
BOARD[ROW][COL]
        ENDIF
    END FOR
END FUNCTION
```

```
START
  INPUT N
  INITIALIZE BOARD[N][N] = 0
  CALL PLACE_QUEENS(1)
END
```

Options:

- A) N-Queens backtracking ☒
- B) Matrix sum
- C) DFS paths
- D) Permutations only

Answer: A) N-Queens backtracking
