

=> MAIN TOPICS

1. BASICS
2. MATHEMATICAL
3. CONDITIONS
4. LOOPS
5. ARRAYS
6. PATTERNS
7. STRINGS
8. NUMBER PROGRAMS
9. NUMBER SYSTEMS
10. FUNCTIONS/METHODS
11. DATE AND TIME
12. OOP CONCEPTS
13. FILE HANDLING
14. EXCEPTION HANDLING
15. THREADING
16. POINTERS
17. MATRIX
18. RECURSION
19. MINI PROJECTS

=> 1. BASICS

1) WRITE A PROGRAM TO DISPLAY “HELLO WORLD” TEXT.

2) WRITE A PROGRAM TO DEMONSTRATE THE USE OF ALL DATA TYPES. (BYTE, INT, SHORT, LONG, FLOAT, DOUBLE, CHAR, STRING ETC.)

3) WRITE A PROGRAM TO SWAP TWO NUMBERS.

4) WRITE A PROGRAM TO SWAP TWO NUMBERS WITHOUT USING THE THIRD VARIABLE.

5) WRITE A PROGRAM TO SWAP TWO STRINGS.

6) WRITE A PROGRAM TO FIND ASCII VALUE OF A CHARACTER.

7) WRITE A PROGRAM TO PRINT ALL ASCII VALUE TABLE.

8) WRITE A PROGRAM TO FIND THE SIZE OF BYTE, INT, SHORT, LONG, FLOAT, DOUBLE, CHAR, STRING ETC.

9) WRITE A PROGRAM TO PRINT YOUR NAME AND AGE.

10) WRITE A PROGRAM TO CALCULATE THE SUM OF TWO NUMBERS.

11) WRITE A PROGRAM TO DISPLAY A MESSAGE BASED ON USER INPUT.

12) WRITE A PROGRAM TO SHOW HOW TO DECLARE AND USE VARIABLES OF DIFFERENT DATA TYPES.

13) WRITE A PROGRAM TO DEMONSTRATE THE USE OF COMMENTS IN CODE.

14) WRITE A PROGRAM TO PRINT YOUR FAVORITE QUOTE.

15) WRITE A PROGRAM TO PRINT ALL DIVISORS OF A NUMBER.

16) WRITE A PROGRAM TO DISPLAY A WELCOME MESSAGE TO THE USER.

17) WRITE A PROGRAM TO TAKE MULTIPLE USER INPUTS (NAMES OR NUMBERS) AND DISPLAY THEM.

18) WRITE A PROGRAM TO SHOW HOW TO DEFINE AND USE CONSTANTS.

19) WRITE A PROGRAM TO TAKE A NUMBER AS INPUT AND PRINT IT IN BINARY FORM.

=> 2. MATHEMATICAL

- 1) WRITE A PROGRAM TO PERFORM ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION, MODULAS OF TWO NUMBERS (ALL DATA TYPES).
- 2) WRITE A PROGRAM TO INPUT FAHRENHEIT AND CELSIUS TEMPERATURE FROM USER AND CONVERT FAHRENHEIT TO CELSIUS AND CELSIUS TO FAHRENHEIT TEMPERATURE AND DISPLAY IT.
- 3) WRITE A PROGRAM TO COMPUTE QUOTIENT AND REMAINDER.
- 4)WRITE A PROGRAM TO PRINT ALL DIVISORS OF A NUMBER.
- 5) WRITE A PROGRAM TO CALCULATE AND PRINT AVERAGE OF 10 NUMBERS ENTERED BY USER.

6) WRITE A PROGRAM TO CALCULATE AND PRINT THE SIMPLE INTEREST ON VALUES ENTERED BY USER.

7) WRITE A PROGRAM TO INPUT A NUMBER FROM USER AND THEN CALCULATE SQUARE AND CUBE OF THAT NUMBER AND DISPLAY ANSWER OF IT.

8) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{2} * b * h$$

$$\text{PERIMETER} = a + b + c$$

9) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN EQUILATERAL TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{4} * \sqrt{3} * a^2$$

$$\text{PERIMETER} = 3a$$

- 10) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN ISOSCELES TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{2} * b * h_b \text{ OR } (b/4) * \text{sqrt}(4a^2 - b^2)$$

$$\text{PERIMETER} = 2a + b$$

- 11) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A SCALENE TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{2} * b * h$$

$$\text{PERIMETER} = a + b + c$$

- 12) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A RIGHT ANGLE TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{2} * b * h$$

$$\text{PERIMETER} = a + b + c$$

13) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN ACUTE ANGLE TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{2} * b * h$$

$$\text{PERIMETER} = a + b + c$$

14) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN OBTUSE ANGLE TRIANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{1}{2} * b * h$$

$$\text{PERIMETER} = a + b + c$$

15) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A SQUARE ON VALUES ENTERED BY USER.

$$\text{AREA} = s^2 \text{ OR } s * s$$

$$\text{PERIMETER} = 4s \text{ OR } 4 * \sqrt{\text{AREA}}$$

16) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A RECTANGLE ON VALUES ENTERED BY USER.

$$\text{AREA} = L * B$$

$$\text{PERIMETER} = 2 * (L + B)$$

17) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A PARALLELOGRAM ON VALUES ENTERED BY USER.

$$\text{AREA} = b * h$$

$$\text{PERIMETER} = 2 * (a + b) \text{ OR } 2 * (s_1 + s_2)$$

18) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A TRAPEZOID ON VALUES ENTERED BY USER.

$$\text{AREA} = ((a_1 + a_2) / 2) * h \text{ OR } (\frac{1}{2}) * h * (b_1 + b_2)$$

$$\text{PERIMETER} = a_1 + a_2 + b_1 + b_2 \text{ OR } s_1 + s_2 + s_3 + s_4$$

19) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A KITE ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{1}{2}\right) * d1 * d2$$

$$\text{PERIMETER} = 2a + 2b \text{ OR } 2 * (s1 + s2)$$

20) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A RHOMBUS ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{1}{2}\right) * d1 * d2$$

$$\text{PERIMETER} = 4a$$

21) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A PENTAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{1}{4}\right) * \sqrt{5(5 + 2\sqrt{5})} * s^2$$

$$\text{PERIMETER} = 5s$$

22) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A HEXAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \frac{3\sqrt{3}}{2} * s^2$$

$$\text{PERIMETER} = 6s$$

23) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A HEPTAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{7}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{7}\right)$$

$$\text{PERIMETER} = 7s$$

24) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN OCTAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = 2 \times s^2 \times (1 + \sqrt{2})$$

$$\text{PERIMETER} = 8s$$

25) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A NONAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{9}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{9}\right)$$

$$\text{PERIMETER} = 9s$$

26) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A DECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{5}{4}\right) \times s^2 \times \sqrt{5 + 2\sqrt{5}}$$

$$\text{PERIMETER} = 10s$$

27) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A HENDECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{11}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{11}\right)$$

$$\text{PERIMETER} = 11s$$

28) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A DODECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = 3 \times s^2 \times (2 + 2\sqrt{3})$$

$$\text{PERIMETER} = 12s$$

29) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A TRIDECAGON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{13}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{13}\right)$$

$$\text{PERIMETER} = 13s$$

30) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A TETRADECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{14}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{14}\right)$$

$$\text{PERIMETER} = 14s$$

- 31) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A PENDEDECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{15}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{15}\right)$$

$$\text{PERIMETER} = 15s$$

- 32) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A HEXDECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{16}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{16}\right)$$

$$\text{PERIMETER} = 16s$$

- 33) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A HEPTDECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{17}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{17}\right)$$

$$\text{PERIMETER} = 17s$$

- 34) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN OCTDECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{18}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{18}\right)$$

$$\text{PERIMETER} = 18s$$

- 35) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN ENNEADECAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{19}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{19}\right)$$

$$\text{PERIMETER} = 19s$$

- 36) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF AN ICOSAGON ON VALUES ENTERED BY USER.

$$\text{AREA} = \left(\frac{20}{4}\right) \times s^2 \times \cot\left(\frac{\pi}{20}\right)$$

$$\text{PERIMETER} = 20s$$

37) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER OF A N-GON ON VALUES ENTERED BY USER.

$$\text{AREA} = (Ns^2)/(4TAN(\pi/N))$$

$$\text{PERIMETER} = Ns$$

38) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND CIRCUMFERENCE OF A CIRCLE ON VALUES ENTERED BY USER.

$$\text{AREA} = \pi r^2$$

$$\text{CIRCUMFERENCE} = 2\pi r$$

39) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND VOLUME AND CIRCUMFERENCE OF A CYLINDER ON VALUES ENTERED BY USER.

$$\text{AREA} = 2\pi r^2 + 2\pi rh$$

$$\text{VOLUME} = \pi r^2 h$$

$$\text{CIRCUMFERENCE} = 2\pi r$$

40) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND VOLUME AND CIRCUMFERENCE OF A CONE ON VALUES ENTERED BY USER.

$$\text{AREA} = \pi r^2 + \pi r s$$

$$\text{VOLUME} = \left(\frac{1}{3}\right) * \pi r^2 h$$

$$\text{CIRCUMFERENCE} = 2\pi r$$

41) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND VOLUME AND PERIMETER OF A CUBE ON VALUES ENTERED BY USER.

$$\text{AREA} = 6s^2$$

$$\text{VOLUME} = s^3$$

$$\text{PERIMETER} = 4s$$

42) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER AND CIRCUMFERENCE OF A CYCLOID ON VALUES ENTERED BY USER.

$$\text{AREA} = 3\pi r^2$$

$$\text{PERIMETER} = 2\pi r$$

$$\text{CIRCUMFERENCE} = 2\pi r$$

- 43) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER AND CIRCUMFERENCE OF AN ELLIPSE ON VALUES ENTERED BY USER.

$$\text{AREA} = \pi ab$$

$$\text{PERIMETER} = 2\pi \sqrt{((a^2 + b^2) / 2)}$$

$$\text{CIRC.} = \pi 3(a + b) - \sqrt{((3a + b)(a + 3b))}$$

- 44) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER AND VOLUME OF A PYRAMID ON VALUES ENTERED BY USER.

AREA OF A RECTANGULAR PYRAMID:

$$A = lw + 2lh + 2wh$$

AREA OF A TRIANGULAR PYRAMID:

$$A = 0.5bh + s_1 + s_2 + s_3$$

AREA OF A POLYGONAL PYRAMID:

$$A = (\frac{1}{2})Pl + B$$

$$\text{PERIMETER} = a_1 + a_2 + a_3 + \dots + a_n$$

$$\text{VOLUME} = (1/3)BH$$

45) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND PERIMETER AND VOLUME OF A PRISM ON VALUES ENTERED BY USER.

AREA OF A PRISM WITH A RECTANGULAR BASE:

$$A = lw$$

AREA OF A PRISM WITH A TRIANGULAR BASE:

$$A = (\frac{1}{2})bh$$

AREA OF A PRISM WITH A POLYGONAL BASE:

$$A = (\frac{1}{2})P_n \cdot a_p$$

$$\text{PERIMETER} = 2L + 2W$$

$$\text{VOLUME} = Bh$$

46) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND VOLUME AND CIRCUMFERENCE OF A SPHERE ON VALUES ENTERED BY USER.

$$\text{AREA} = 4\pi r^2$$

$$\text{VOLUME} = (\frac{4}{3})\pi r^3$$

$$\text{CIRCUMFERENCE} = 2\pi r$$

47) WRITE A PROGRAM TO CALCULATE AND PRINT THE AREA AND VOLUME AND CIRCUMFERENCE OF A HEMISPHERE ON VALUES ENTERED BY USER.

$$\text{AREA} = 2\pi r^2$$

$$\text{VOLUME} = (2/3) \pi r^3$$

$$\text{CIRCUMFERENCE} = 2\pi r$$

48) WRITE A PROGRAM TO PRINT MULTIPLICATION TABLE OF A GIVEN NUMBER BY USER WITHOUT USING LOOPS.

49) WRITE A PROGRAM TO INPUT MARKS OF ALL SUBJECTS OF A STUDENT FROM USER AND CALCULATE TOTAL MARKS AND AVERAGE AND PERCENTAGE OF MARKS AND THEN DISPLAY IT.

50) WRITE A PROGRAM TO CALCULATE DISTANCE BETWEEN TWO POINTS TAKING INPUT FROM THE USER (PYTHAGOREAN THEOREM).

- 51) WRITE A PROGRAM TO ENTER SECONDS FROM USER AND CONVERT SECONDS INTO HOURS AND MINUTES AND SECONDS AND THEN DISPLAY IT IN ALL DATE FORMATS.
- 52) WRITE A PROGRAM TO ENTER DISTANCE INTO KILOMETER FROM USER AND CONVERT IT INTO METER, FEET, INCHES, CENTIMETER THEN DISPLAY IT.
- 53) WRITE A PROGRAM TO CALCULATE GROSS SALARY OF A PERSON ON VALUES ENTERED BY USER AND THEN DISPLAY IT.
- 54) WRITE A PROGRAM TO CHECK WHETHER THE ENTERED NUMBER BY USER CAN BE EXPRESSED AS A SUM OF TWO PRIME NUMBERS.
- 55) WRITE A PROGRAM TO PERFORM ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION OF TWO COMPLEX NUMBERS ENTERED BY USER.

56) WRITE A PROGRAM TO EVALUATE THE GIVEN POLYNOMIAL EQUATION BY USER.

57) WRITE A PROGRAM TO FIND EXPONENT POWER SERIES.

58) WRITE A PROGRAM TO FIND OUT THE ROOTS OF A QUADRATIC EQUATION.

59) WRITE A PROGRAM TO INPUT TWO BINARY NUMBERS FROM USER AND THEN PERFORM ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION OF THAT TWO BINARY NUMBERS AND THEN DISPLAY ANSWER OF IT.

60) WRITE A PROGRAM TO MULTIPLY GIVEN NUMBER BY 4 USING BITWISE OPERATORS.

→ The bitwise operators are or, and, xor, not, left shift, right shift. Program uses left shift operator for this.

61) WRITE A PROGRAM TO READ THE VALUES OF X, Y AND Z FROM USER AND PRINT THE RESULTS EXPRESSIONS IN ONE LINE.

$$\rightarrow 1 \ (x + y + z) / (x-y-z)$$

$$2 \ (x + y + z) / 3$$

$$3 \ (x + y) * (x-y) * (y-z)$$

62) WRITE A PROGRAM TO SOLVE SECOND ORDER QUADRATIC EQUATION.

63) WRITE A PROGRAM TO SWAP THE CONTENTS OF TWO NUMBERS USING BITWISE XOR OPERATION.

64) WRITE A PROGRAM TO CALCULATE ARCTANGENT, 2 PARAMETERS: HOW TO USE ATAN2.

65) WRITE A PROGRAM TO CALCULATE LOGARITHM BASE 10: HOW TO USE LOG10.

- 66) WRITE A PROGRAM TO CHECK A CIRCLE THROUGH THREE POINTS.
- 67) WRITE A PROGRAM THAT COMPUTES THE ACCURACY OF THE FLOATING POINT NUMBERS IN STORAGE AND CALCULATIONS.
- 68) WRITE A PROGRAM TO FIND LCM BY CALCULATION BY FINDING GCD.

→ The LCM of two integers j_1 and j_2 is the smallest positive integer that is perfectly divisible by both j_1 and j_2 (without a remainder).

For example: the LCM of 72 and 120 is 360.

- 69) WRITE A PROGRAM TO FIND THE GCD OF TWO NUMBERS USING EUCLID'S ALGORITHM.
- 70) WRITE A PROGRAM TO FIND THE STANDARD FORM OF A QUADRATIC EQUATION.

→ The standard form of a quadratic equation is:

$$ax^2 + bx + c = 0, \text{ where}$$

a, b and c are real numbers and

$$a \neq 0$$

The term $b^2 - 4ac$ is known as the determinant of a quadratic equation. The determinant tells the nature of the roots.

If determinant is greater than 0, the roots are real and different.

If determinant is equal to 0, the roots are real and equal.

If determinant is less than 0, the roots are complex and different.

71) WRITE A PROGRAM TO FIND THE SUM OF A.P SERIES.

→ This program is used to find the sum of the arithmetic progression series. Here A.P stands for arithmetic progression. A sequence of terms each of which, after the first, is derived

by adding to the preceding one a common difference: 5, 9, 13, 17, etc. forms an arithmetic progression. The formula used in this program are $l = a + (n - 1)d$ where l is the last term of a finite sequence. $S_n = \frac{n}{2}(2a + (n-1)d)$ where S_n is the sum of n terms.

72) WRITE A PROGRAM TO FIND THE SUM OF G.P SERIES.

→ This program is used to find the sum of the geometric progression series. Here G.P stands for geometric progression. A geometric progression, or GP, is a sequence where each new term after the first is obtained by multiplying the preceding term by a constant r , called the common ratio. The formula used in this program are $T_n = a * (r ^ (n - 1))$. where T_n is the last term of a finite sequence. $S_n = \frac{a(1 - r ^ n + 1)}{(1 - r)}$ where S_n is the sum of n terms.

73) WRITE A PROGRAM TO FIND THE SUM OF H.P SERIES.

→ This program is used to find the sum of the harmonic progression series. Here H.P stands for harmonic progression. Harmonic progression is a progression formed by taking the reciprocals of an arithmetic progression.

74) WRITE A PROGRAM TO CALCULATE AND PRINT THE VALUE OF nCr (COMBINATION OR SELECTION) ON VALUES ENTERED BY USER.

FORMULA:

$$C(n,r) = n! / (r!(n-r)!)$$

For $0 \leq r \leq n$

75) WRITE A PROGRAM TO CALCULATE AND PRINT THE VALUE OF nPr (PERMUTATION OR ARRANGEMENT) ON VALUES ENTERED BY USER.

FORMULA:

$$P(n,r) = n! / (n-r)!$$

For $0 \leq r \leq n$

76) WRITE A PROGRAM TO CALCULATE THE MEAN, MEDIAN, MODE ON THE DATA ENTERED BY USER. (USE ALL FORMULAS)

FORMULAS:

=>MEAN

→**Mean of Ungrouped Data:**

$$\bar{x} = \frac{\sum x_i}{n}$$

→**Mean of Discrete Grouped Data:**

$$\bar{x} = \frac{\sum f_i x_i}{n}$$

Mean by assumed mean method:

$$\bar{x} = A + \frac{\sum f_i d_i}{n};$$

Where $d_i = x_i - A$

Here, A can be any value of x_i .

→**Mean of Continuous Grouped Data:**

$\bar{x} = \frac{\sum f_i x_i}{n}$; where x_i = Mid value of the respective class.

Mean by assumed mean method:

$$\bar{x} = A + \frac{\sum f_i d_i}{n};$$

Where $d_i = x_i - A$

Mean by step deviation method:

$$\bar{x} = A + \frac{\sum f_i u_i}{n} * C;$$

Where $u_i = (x_i - A)/C$

Here, A can be any value of x_i .

C is the class length.

=>MEDIAN

→Median of Ungrouped Data & Discrete Grouped Data:

- If n is odd number, then

$$M = \left(\frac{n+1}{2}\right)^{th} \text{ observation}$$

- If n is even number, then

$$M = \frac{\left(\frac{n}{2}\right)^{th} \text{ observation} + \left(\frac{n}{2}+1\right)^{th} \text{ observation}}{2}$$

→Median of Continuous Grouped Data:

$$M = L + \left(\frac{\frac{n}{2} - F}{f} \right) \times C$$

Where,

- Median Class = Class whose cumulative frequency with property $\min \{ cf \mid cf \geq n/2 \}$.
- L = Lower boundary point of the median class.
- n = Total number of observation (sum of the frequencies).
- F = Cumulative frequency of the class preceding the median class.
- f = The frequency of the median class.
- C = Class length.

=>MODE

→Mode of Ungrouped Data: Most repeated observation among given data is called mode of ungrouped data.

→ Mode of Discrete Frequency

Distribution: The value of observation with highest frequency.

→ Mode of Continuous Frequency

Distribution:

$$z = L + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2} \right) \times C$$

Where,

- Modal Class = A class with highest frequency.
- L = Lower boundary point of the modal class.
- C = Class length.
- f_1 = Frequency of the modal class.
- f_0 = Frequency of the class preceding the modal class.
- f_2 = Frequency of the class succeeding the modal class.

=> 3. CONDITIONS

- 1) WRITE A PROGRAM TO DISPLAY “HELLO WORLD” TEXT.
- 2) WRITE A PROGRAM TO CHECK IF A NUMBER IS EVEN OR ODD.
- 3) WRITE A PROGRAM TO CHECK IF A NUMBER IS POSITIVE, NEGATIVE, OR ZERO.
- 4) WRITE A PROGRAM TO CHECK IF A NUMBER IS PRIME OR NOT.
- 5) WRITE A PROGRAM TO CHECK FOR ALPHANUMERIC CHARACTERS IN A STRING.
- 6) WRITE A PROGRAM TO DETERMINE THE GRADE OF A STUDENT BASED ON MARKS USING IF-ELSE.

- 7) WRITE A PROGRAM TO FIND THE LARGEST OF THREE NUMBERS USING IF-ELSE.
- 8) WRITE A PROGRAM TO DISPLAY A MESSAGE BASED ON A USER'S AGE (e.g., Child, Teen, Adult).
- 9) WRITE A PROGRAM TO CHECK IF A YEAR IS A LEAP YEAR OR NOT.
- 10) WRITE A PROGRAM TO CHECK THE RANGE OF A NUMBER (e.g., 1-100).
- 11) WRITE A PROGRAM TO ASSIGN LETTER GRADE (A, B, C, D, F) BASED ON A SCORE USING IF-ELSE.
- 12) WRITE A PROGRAM TO DETERMINE THE TYPE OF TRIANGLE (Scalene, Isosceles, Equilateral) BASED ON SIDE LENGTHS.

- 13) WRITE A PROGRAM TO CHECK IF A NUMBER IS DIVISIBLE BY 5 AND 10.
- 14) WRITE A PROGRAM TO FIND THE SMALLEST OF THREE NUMBERS USING IF-ELSE.
- 15) WRITE A PROGRAM TO CHECK IF A STRING IS PALINDROME OR NOT USING CONDITIONS.
- 16) WRITE A PROGRAM TO VALIDATE A PASSWORD (e.g., length, special character).
- 17) WRITE A PROGRAM TO FIND OUT IF A PERSON IS ELIGIBLE TO VOTE BASED ON AGE.
- 18) WRITE A PROGRAM TO DISPLAY THE RESULT OF A STUDENT BASED ON TOTAL MARKS (Pass/Fail).

- 19) WRITE A PROGRAM TO FIND THE ASCII VALUE OF A CHARACTER AND DISPLAY A MESSAGE BASED ON IT (e.g., Lowercase, Uppercase).
- 20) WRITE A PROGRAM TO CHECK IF A NUMBER IS IN A SPECIFIED RANGE (e.g., 50-100).
- 21) WRITE A PROGRAM TO CHECK IF A TRIANGLE IS VALID BASED ON SIDE LENGTHS.
- 22) WRITE A PROGRAM TO FIND THE SMALLEST OF FOUR NUMBERS USING IF-ELSE.
- 23) WRITE A PROGRAM TO DISPLAY A MESSAGE IF A USER ENTERS A SPECIFIC KEYWORD.
- 24) WRITE A PROGRAM TO DISPLAY THE TYPE OF ANGLE (Acute, Right, Obtuse) BASED ON USER INPUT.

- 25) WRITE A PROGRAM TO CHECK IF A YEAR IS A CENTURY YEAR.
- 26) WRITE A PROGRAM TO DISPLAY THE RESULT OF A RACE BASED ON FINISHING TIMES.
- 27) WRITE A PROGRAM TO CHECK IF A NUMBER IS A POWER OF TWO.
- 28) WRITE A PROGRAM TO CHECK IF A NUMBER IS A NON-NEGATIVE INTEGER.
- 29) WRITE A PROGRAM TO CHECK IF A NUMBER IS A CUBE NUMBER.
- 30) WRITE A PROGRAM TO DETERMINE IF A GIVEN DATE FALLS IN WINTER, SPRING, SUMMER, OR FALL.
- 31) WRITE A PROGRAM TO DISPLAY AN EMOJI BASED ON USER MOOD INPUT (e.g., Happy, Sad, Angry).

- 32) WRITE A PROGRAM TO CHECK IF A NUMBER IS DIVISIBLE BY 3 OR 7 BUT NOT BOTH.
- 33) WRITE A PROGRAM TO DETERMINE IF A GIVEN TIME IS AM OR PM.
- 34) WRITE A PROGRAM TO CHECK IF TWO CHARACTERS ARE ALPHABETICALLY CONSECUTIVE.
- 35) WRITE A PROGRAM TO CHECK IF A NUMBER IS A DOUBLE DIGIT NUMBER.
- 36) WRITE A PROGRAM TO DETERMINE IF A GIVEN TIME IS IN THE PAST, PRESENT, OR FUTURE.
- 37) WRITE A PROGRAM TO CHECK IF A NUMBER IS EXACTLY DIVISIBLE BY ANOTHER NUMBER GIVEN BY THE USER.

- 38) WRITE A PROGRAM TO CHECK IF TWO GIVEN NUMBERS ARE EQUAL, GREATER, OR LESS THAN EACH OTHER.
- 39) WRITE A PROGRAM TO CHECK IF A GIVEN NUMBER IS A TWO-DIGIT PRIME NUMBER.
- 40) WRITE A PROGRAM TO CHECK IF A NUMBER IS ODD OR EVEN WITHOUT USING MODULO OPERATOR.
- 41) WRITE A PROGRAM TO DETERMINE IF A GIVEN YEAR IS THE START OF A NEW DECADE.
- 42) WRITE A PROGRAM TO CHECK IF A NUMBER IS WITHIN A SPECIFIED RANGE BUT EXCLUDES BOTH BOUNDARIES (e.g., $10 < x < 50$).
- 43) WRITE A PROGRAM TO CHECK IF A NAME STARTS AND ENDS WITH THE SAME LETTER.

- 44) WRITE A PROGRAM TO CHECK IF A NUMBER IS AN INTEGER OR FLOAT WITHOUT USING LOOPS.
- 45) WRITE A PROGRAM TO IMPLEMENT A SIMPLE ATM FUNCTIONALITY (WITH BALANCE CHECK).
- 46) WRITE A PROGRAM TO DISPLAY THE NAME OF A DAY BASED ON A NUMBER INPUT (1-7) USING SWITCH CASE.
- 47) WRITE A PROGRAM TO DISPLAY THE MONTH NAME BASED ON A NUMBER INPUT (1-12) USING SWITCH CASE.
- 48) WRITE A PROGRAM TO DISPLAY A GRADE BASED ON A USER'S INPUT SCORE USING SWITCH CASE.
- 49) WRITE A PROGRAM TO DISPLAY THE NUMBER OF DAYS IN A MONTH BASED ON MONTH INPUT (1-12) USING SWITCH CASE.

- 50) WRITE A PROGRAM TO DISPLAY THE SEASON BASED ON MONTH NUMBER USING SWITCH CASE (e.g., 12 = Winter).
- 51) WRITE A PROGRAM TO DISPLAY COURSE TYPE BASED ON STUDENT'S INPUT CODE USING SWITCH CASE (e.g., 101 -> Math, 102 -> Science).
- 52) WRITE A PROGRAM TO DISPLAY THE CATEGORY OF A BOOK BASED ON THE FIRST LETTER OF ITS ISBN CODE USING SWITCH CASE.
- 53) WRITE A PROGRAM TO DISPLAY THE NAME OF A PLANET BASED ON ITS ORDER FROM THE SUN USING SWITCH CASE (e.g., 1 -> Mercury, 2 -> Venus).
- 54) WRITE A PROGRAM TO DISPLAY CURRENCY NAME BASED ON CURRENCY SYMBOL INPUT USING SWITCH CASE (e.g., \$ -> Dollar, € -> Euro).

- 55) WRITE A PROGRAM TO DISPLAY SPORTS CATEGORY BASED ON A USER'S INPUT USING SWITCH CASE (e.g., 1 -> Indoor, 2 -> Outdoor).
- 56) WRITE A PROGRAM TO DISPLAY A USER'S ACCESS LEVEL BASED ON USER ROLE CODE INPUT USING SWITCH CASE (e.g., 1 -> Admin, 2 -> Guest).
- 57) WRITE A PROGRAM TO DISPLAY TRAFFIC LIGHT COLOR MEANING BASED ON COLOR INPUT USING SWITCH CASE (e.g., Red -> Stop, Green -> Go).
- 58) WRITE A PROGRAM TO DISPLAY CLASS TIME BASED ON DAY OF WEEK INPUT USING SWITCH CASE (e.g., 1 -> Monday, 2 -> Tuesday).
- 59) WRITE A PROGRAM TO DISPLAY COFFEE SIZE BASED ON USER INPUT (e.g., S -> Small, M -> Medium, L -> Large) USING SWITCH CASE.

- 60) WRITE A PROGRAM TO DISPLAY THE TYPE OF ACCOUNT BASED ON CODE (e.g., S -> Savings, C -> Checking) USING SWITCH CASE.
- 61) WRITE A PROGRAM TO DISPLAY THE SHAPE BASED ON NUMBER OF SIDES INPUT USING SWITCH CASE (e.g., 3 -> Triangle, 4 -> Square).
- 62) WRITE A PROGRAM TO DISPLAY THE COURSE FEE BASED ON PROGRAM TYPE INPUT USING SWITCH CASE (e.g., 1 -> Undergraduate, 2 -> Postgraduate).
- 63) WRITE A PROGRAM TO DISPLAY WORK SHIFT BASED ON TIME RANGE INPUT USING SWITCH CASE.
- 64) WRITE A PROGRAM TO DISPLAY THE CATEGORY OF ARTWORK BASED ON ARTIST'S NAME INPUT USING SWITCH CASE.

- 65) WRITE A PROGRAM TO DISPLAY PIZZA SIZE BASED ON USER SELECTION (e.g., S -> Small, M -> Medium) USING SWITCH CASE.
- 66) WRITE A PROGRAM TO DISPLAY THE GENRE OF A BOOK BASED ON INITIAL LETTER USING SWITCH CASE.
- 67) WRITE A PROGRAM TO DISPLAY THE TYPE OF PAYMENT METHOD BASED ON CODE (e.g., C -> Credit, D -> Debit) USING SWITCH CASE.
- 68) WRITE A PROGRAM TO DISPLAY OPERATING SYSTEM NAME BASED ON SHORT CODE INPUT (e.g., WIN -> Windows, MAC -> MacOS) USING SWITCH CASE.
- 69) WRITE A PROGRAM TO DISPLAY ELEMENT NAME BASED ON SYMBOL

=> 4. LOOPS

- 1) WRITE A PROGRAM TO PRINT 0 TO 10 USING ALL THE LOOPS.
- 2) WRITE A PROGRAM TO PRINT 10 TO 0 USING ALL THE LOOPS.
- 3) WRITE A PROGRAM TO PRINT 0 TO N USING ALL THE LOOPS.
- 4) WRITE A PROGRAM TO PRINT N TO 0 USING ALL THE LOOPS.
- 5) WRITE A PROGRAM TO PRINT NUMBERS BETWEEN TWO GIVEN RANGE BY THE USER USING ALL THE LOOPS (FOR EXAMPLE: [A=1, B=10] THEN PRINT FROM A TO B AND FROM B TO A).
- 6) WRITE A PROGRAM TO COUNT ODD AND EVEN NUMBERS FROM USER GIVEN RANGE USING ALL THE LOOPS.

7) WRITE A PROGRAM TO COUNT POSITIVE AND NEGATIVE NUMBERS FROM USER GIVEN RANGE USING ALL THE LOOPS.

8) WRITE A PROGRAM TO PERFORM THE ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION OF ALL THE NUMBERS FROM USER GIVEN RANGE USING ALL THE LOOPS.

INPUT:

ENTER NUMBER1: 1

ENTER NUMBER2: 5

OUTPUT:

ADDITION: 15

SUBTRACTION: -13

MULTIPLICATION: 120

DIVISION: 0.0083

9) WRITE A PROGRAM TO COUNT THE TOTAL NUMBERS OF DIGITS IN A NUMBER GIVEN BY USER USING ALL THE LOOPS.

- 10) WRITE A PROGRAM TO COUNT NUMBERS WHICH ARE DIVISIBLE BY 1,2,5,9,10 FROM USER GIVEN USING ALL THE LOOPS.
- 11) WRITE A PROGRAM TO INPUT A NUMBER FROM USER AND GIVE FACTORIAL OF THAT NUMBER USING ALL THE LOOPS.
- 12) WRITE A PROGRAM TO INPUT TWO NUMBERS FROM USER AND PRINT FIRST RAISE TO SECOND NUMBER USING ALL THE LOOPS.
- 13) WRITE A PROGRAM TO INPUT 10 NUMBERS FROM USER WITHIN LOOP AND GIVE SUM OF ALL THE NUMBERS USING ALL THE LOOPS.
- 14) WRITE A PROGRAM TO INPUT 10 NUMBERS FROM USER WITHIN LOOP AND GIVE SUM AND COUNT OF ODD AND EVEN NUMBERS, AND ALSO GIVE SUM AND COUNT OF POSITIVE AND NEGATIVE NUMBERS USING ALL THE LOOPS.

- 15) WRITE A PROGRAM TO INPUT 10 NUMBERS FROM USER WITHIN LOOP AND GIVE GREATEST AND SMALLEST NUMBER FROM THEM USING ALL THE LOOPS.
- 16) WRITE A PROGRAM TO INPUT A NUMBER FROM USER AND PRINT IT'S MULTIPLICATION TABLE USING ALL THE LOOPS.
- 17) WRITE A PROGRAM TO INPUT 3 NUMBERS FROM USER (1) FIRST NUMBER: TO PRINT TABLE OF THIS NUMBER, (2) SECOND NUMBER: STARTING RANGE, (3) THIRD NUMBER: ENDING RANGE USING ALL THE LOOPS.
- 18) WRITE A PROGRAM TO INPUT TWO VALUES FROM USER AND GIVE FACTORIAL OF THE RANGE WITHOUT USING THE NESTED LOOP.

EXAMPLE

INPUT:

ENTER NUM1: 3

ENTER NUM2: 6

OUTPUT:

$3!=6$

$4!=24$

$5!=120$

$6!=720$

19) WRITE A PROGRAM TO INPUT A NUMBER FROM USER AND PRINT ALL FACTORS OF THAT NUMBER USING ALL THE LOOPS.

20) WRITE A PROGRAM TO INPUT NUMBERS FROM USER. USER MAY INPUT POSITIVE NUMBERS AND NEGATIVE NUMBERS MAKE PROGRAM TO COUNT POSITIVE AND NEGATIVE NUMBERS SEPARATELY. THERE IS NO LIMIT TO INPUT NUMBERS. INPUT MUST BE STOPPED WHEN USER WILL INPUT ZERO.

EXAMPLE 1:

INPUT:

ENTER A NUMBER: 10
ENTER A NUMBER: -10
ENTER A NUMBER: 1
ENTER A NUMBER: 0

OUTPUT:

NEGATIVE: 1
POSITIVE: 2

EXAMPLE 2:

INPUT:

ENTER A NUMBER: 10
ENTER A NUMBER: -10
ENTER A NUMBER: 1
ENTER A NUMBER: 10
ENTER A NUMBER: -10
ENTER A NUMBER: -2
ENTER A NUMBER: 0

OUTPUT:

NEGATIVE: 3
POSITIVE: 3

21) WRITE A MENU DRIVEN PROGRAM
FOR RESTAURENT AS FOLLOWING:

1.PIZZA	100
2.HOT DOG	80
3.BURGER	70
4.SANDWICH	50
5.FRENCH FRIES	30
6.EXIT	

EXAMPLE 1:

INPUT:

ENTER YOUR CHOICE: 1

ENTER QUANTITY: 2

ENTER YOUR CHOICE: 5

ENTER QUANTITY: 1

ENTER YOUR CHOICE: 6

YOUR BILL IS: 230

EXAMPLE 2:

INPUT:

ENTER YOUR CHOICE: 5

ENTER QUANTITY: 1

ENTER YOUR CHOICE: 1

ENTER QUANTITY: 3

ENTER YOUR CHOICE: 2

ENTER QUANTITY: 2

ENTER YOUR CHOICE: 3

ENTER QUANTITY: 1

ENTER YOUR CHOICE: 6

YOUR BILL IS: 56

22) WRITE A PROGRAM TO INPUT TWO NUMBERS FROM USER AND FIND HCF AND LCM OF THOSE TWO NUMBERS USING ALL THE LOOPS.

EXAMPLE:

INPUT:

ENTER THE NUMBER:15

ENTER THE NUMBER:25

OUTPUT:

HCF:5

LCM:75

23) WRITE A PROGRAM TO INPUT A SINGLE NUMBER FROM USER AND PRINT EACH DIGIT IN SEPARATE LINE USING ALL THE LOOPS.

EXAMPLE:

INPUT:

ENTER THE NUMBER:157

OUTPUT:

1

5

7

24) WRITE A PROGRAM TO INPUT A SINGLE NUMBER FROM USER AND PRINT EACH DIGIT IN SEPARATE LINE IN REVERSE ORDER USING ALL THE LOOPS.

EXAMPLE:

INPUT:

ENTER THE NUMBER:157

OUTPUT:

7

5

1

25) WRITE A PROGRAM TO INPUT A SINGLE NUMBER FROM USER AND PRINT SUM OF EACH DIGIT.

EXAMPLE:

INPUT:

ENTER THE NUMBER:157

OUTPUT:

SUM=13

26) WRITE A PROGRAM TO INPUT A SINGLE NUMBER FROM USER AND COUNT TOTAL NUMBER OF DIGITS IN THAT NUMBER AND ALSO COUNT HOW MANY ARE ODD AND HOW MANY ARE EVEN DIGITS.

EXAMPLE:

INPUT:

ENTER THE NUMBER:157

OUTPUT:

TOTAL DIGITS: 3

ODD: 3

EVEN: 0

27) WRITE A PROGRAM TO INPUT TWO NUMBERS FROM USER AND THEN COUNT TOTAL NUMBER OF TIME SECOND NUMBER IS PRESENT IN FIRST NUMBER AND THEN DISPLAY THE TOTAL OF IT.

EXAMPLE:

INPUT:

ENTER THE FIRST NUMBER: 1213141516

ENTER THE SECOND NUMBER:1

OUTPUT:

TOTAL NUMBER OF 1 IS: 5

28) WRITE A PROGRAM TO INPUT A
SINGLE NUMBER FROM USER AND PRINT
SMALLEST AND LARGEST DIGIT.

EXAMPLE:

INPUT:

ENTER THE NUMBER:157

OUTPUT:

SMALLEST: 1

LARGEST: 7

29) WRITE A PROGRAM TO CREATE A CALCULATOR AND INPUT OPERANDS (NUMBERS) AND OPERATORS FROM USER AND THEN CALCULATE ANSWER AND DISPLAY IT AND THEN ASK USER TO PERFORM OPERATION UNTIL USER SAYS NO (CHOICE = 0).

EXAMPLE:

ANSWER = 0

ENTER FIRST NUMBER: 1

ENTER SECOND NUMBER: 2

ENTER OPERATOR: +

ANSWER = 3

DO YOU WANT TO CONTINUE? (0 FOR EXIT /1 TO CONTINUE)

ENTER YOUR CHOICE: 1

ANSWER = 0

ENTER FIRST NUMBER: 2

ENTER SECOND NUMBER: 1

ENTER OPERATOR: -

ANSWER = 1

DO YOU WANT TO CONTINUE? (0 FOR
EXIT /1 TO CONTINUE)

ENTER YOUR CHOICE: 0

30) WRITE A PROGRAM TO COUNT THE
NUMBER OF WORDS IN A SENTENCE.

31) WRITE A PROGRAM TO CREATE A
SIMPLE PASSWORD VALIDATION
(CHECKING LENGTH).

32) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1+2+3+4+5+\dots+N$$

33) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1+3+5+7+9+\dots+N$$

34) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2+4+6+8+10+\dots+N$$

35) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1!+2!+3!+4!+5!+\dots+N!$$

36) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1!+3!+5!+7!+9!+\dots+N!$$

37) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2!+4!+6!+8!+10!+\dots+N!$$

38) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^2+2^2+3^2+4^2+5^2+\dots+N^2$$

39) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^2+3^2+5^2+7^2+9^2+\dots+N^2$$

40) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2^2+4^2+6^2+8^2+10^2+\dots+N^2$$

41) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^3+2^3+3^3+4^3+5^3+\dots+N^3$$

42) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^3+3^3+5^3+7^3+9^3+\dots+N^3$$

43) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2^3+4^3+6^3+8^3+10^3+\dots+N^3$$

44) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^n+2^n+3^n+4^n+5^n+\dots+N^n$$

45) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^n+3^n+5^n+7^n+9^n+\dots+N^n$$

46) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2^n+4^n+6^n+8^n+10^n+\dots+N^n$$

47) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1+4+9+16+25+\dots+N$$

48) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1+8+27+64+125+\dots+N$$

49) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1/1!+2/2!+3/3!+4/4!+5/5!+\dots+N/N!$$

50) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1/1!+3/3!+5/5!+7/7!+9/9!+\dots+N/N!$$

51) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2/2!+4/4!+6/6!+8/8!+10/10!+\dots+N/N!$$

52) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^1+2^3+3^5+4^7+5^9+\dots+N^n$$

53) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^1+3^3+5^5+7^7+9^9+\dots+N^n$$

54) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2^1+4^3+6^5+8^7+10^9+\dots+N^n$$

55) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^2+2^4+3^6+4^8+5^{10}+\dots+N^n$$

56) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1^2+3^4+5^6+7^8+9^{10}+\dots+N^n$$

57) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=2^2+4^4+6^6+8^8+10^{10}+\dots+N^n$$

58) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1-2+3-4+5-6+7-8+9-10+\dots+(N-1)-N$$

59) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=1+2-3+4-5+6-7+8-9+10-\dots-(N-1)+N$$

60) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=a^1+a^2+a^3+a^4+a^5+\dots+a^N$$

61) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=a^1+a^3+a^5+a^7+a^9+\dots+a^N$$

62) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=a^2+a^4+a^6+a^8+a^{10}+\dots+a^N$$

63) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=a^n+a^n+a^n+a^n+a^n+\dots+a^n$$

64) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(1*2)+(2*3)+(3*4)+(4*5)+(5*6)+\dots+N$$

65) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(1*2)+(3*4)+(5*6)+(7*8)+(9*10)+\dots+N$$

66) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(1*3)+(3*5)+(5*7)+(7*9)+(9*11)+\dots+N$$

67) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(1*3)+(5*7)+(9*11)+(13*15)+(17*9)+\dots+N$$

68) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(2*4)+(4*6)+(6*8)+(8*10)+(10*12)+\dots+N$$

69) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(2*4)+(6*8)+(10*12)+(14*16)+(18*20)+\dots+N$$

70) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1+(1/2)+(1/3)+(1/4)+\dots+(1/N)$

71) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1-(1/2)-(1/3)-(1/4)-\dots-(1/N)$

72) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1*(1/2)*(1/3)*(1/4)*\dots*(1/N)$

73) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1/(1/2)/(1/3)/(1/4)/\dots/(1/N)$

74) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1+(1/1)+(1/3)+(1/5)+\dots+(1/N)$

75) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1-(1/1)-(1/3)-(1/5)-\dots-(1/N)$

76) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1*(1/1)*(1/3)*(1/5)*\dots*(1/N)$

77) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1/(1/1)/(1/3)/(1/5)/.../(1/N)$

78) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1+(1/2)+(1/4)+(1/6)+...+(1/N)$

79) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1-(1/2)-(1/4)-(1/6)-...-(1/N)$

80) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1*(1/2)*(1/4)*(1/6)*...*(1/N)$

81) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1/(1/2)/(1/4)/(1/6)/.../(1/N)$

82) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1+(1+2)/(1*2)+(1+2+3)/(1*2*3)+...+N$

83) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=1+(1-2)/(1/2)+(1-2-3)/(1/2/3)+...+N$

- 84) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=(a+1)+(a+2)+(a+3)+(a+4)+(a+5)+\dots+(a+N)$
- 85) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=(a+1)+(a+3)+(a+5)+(a+7)+(a+9)+\dots+(a+N)$
- 86) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=(a+2)+(a+4)+(a+6)+(a+8)+(a+10)+\dots+(a+N)$
- 87) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=(a-1)+(a-2)+(a-3)+(a-4)+(a-5)+\dots+(a-N)$
- 88) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=(a-1)+(a-3)+(a-5)+(a-7)+(a-9)+\dots+(a-N)$
- 89) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
 $S=(a-2)+(a-4)+(a-6)+(a-8)+(a-10)+\dots+(a-N)$

90) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*1)+(a*2)+(a*3)+(a*4)+(a*5)+\dots+(a*N)$$

91) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*1)+(a*3)+(a*5)+(a*7)+(a*9)+\dots+(a*N)$$

92) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*2)+(a*4)+(a*6)+(a*8)+(a*10)+\dots+(a*N)$$

93) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)+(a/2)+(a/3)+(a/4)+(a/5)+\dots+(a/N)$$

94) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)+(a/3)+(a/5)+(a/7)+(a/9)+\dots+(a/N)$$

95) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/2)+(a/4)+(a/6)+(a/8)+(a/10)+\dots+(a/N)$$

96) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+1)-(a+2)-(a+3)-(a+4)-(a+5)-\dots-(a+N)$$

97) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+1)-(a+3)-(a+5)-(a+7)-(a+9)-\dots-(a+N)$$

98) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+2)-(a+4)-(a+6)-(a+8)-(a+10)-\dots-(a+N)$$

99) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-1)-(a-2)-(a-3)-(a-4)-(a-5)-\dots-(a-N)$$

100) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-1)-(a-3)-(a-5)-(a-7)-(a-9)-\dots-(a-N)$$

101) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-2)-(a-4)-(a-6)-(a-8)-(a-10)-\dots-(a-N)$$

102) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*1)-(a*2)-(a*3)-(a*4)-(a*5)-\dots-(a*N)$$

103) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*1)-(a*3)-(a*5)-(a*7)-(a*9)-\dots-(a*N)$$

104) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*2)-(a*4)-(a*6)-(a*8)-(a*10)-\dots-(a*N)$$

105) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)-(a/2)-(a/3)-(a/4)-(a/5)-\dots-(a/N)$$

106) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)-(a/3)-(a/5)-(a/7)-(a/9)-\dots-(a/N)$$

107) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/2)-(a/4)-(a/6)-(a/8)-(a/10)-\dots-(a/N)$$

108) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+1)*(a+2)*(a+3)*(a+4)*(a+5)*\dots*(a+N)$$

109) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+1)*(a+3)*(a+5)*(a+7)*(a+9)*\dots*(a+N)$$

110) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+2)*(a+4)*(a+6)*(a+8)*(a+10)*\dots*(a+N)$$

111) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-1)*(a-2)*(a-3)*(a-4)*(a-5)*\dots*(a-N)$$

112) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-1)*(a-3)*(a-5)*(a-7)*(a-9)*\dots*(a-N)$$

113) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-2)*(a-4)*(a-6)*(a-8)*(a-10)*\dots*(a-N)$$

114) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*1)*(a*2)*(a*3)*(a*4)*(a*5)*...*(a*N)$$

115) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*1)*(a*3)*(a*5)*(a*7)*(a*9)*...*(a*N)$$

116) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a*2)*(a*4)*(a*6)*(a*8)*(a*10)*...*(a*N)$$

117) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)*(a/2)*(a/3)*(a/4)*(a/5)*...*(a/N)$$

118) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)*(a/3)*(a/5)*(a/7)*(a/9)*...*(a/N)$$

119) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/2)*(a/4)*(a/6)*(a/8)*(a/10)*...*(a/N)$$

120) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+1)/(a+2)/(a+3)/(a+4)/(a+5)/.../(a+N)$$

121) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+1)/(a+3)/(a+5)/(a+7)/(a+9)/.../(a+N)$$

122) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a+2)/(a+4)/(a+6)/(a+8)/(a+10)/.../(a+N)$$

123) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-1)/(a-2)/(a-3)/(a-4)/(a-5)/.../(a-N)$$

124) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-1)/(a-3)/(a-5)/(a-7)/(a-9)/.../(a-N)$$

125) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a-2)/(a-4)/(a-6)/(a-8)/(a-10)/.../(a-N)$$

126) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a^1)/(a^2)/(a^3)/(a^4)/(a^5)/.../(a^N)$$

127) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a^1)/(a^3)/(a^5)/(a^7)/(a^9)/.../(a^N)$$

128) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a^2)/(a^4)/(a^6)/(a^8)/(a^{10})/.../(a^N)$$

129) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)/(a/2)/(a/3)/(a/4)/(a/5)/.../(a/N)$$

130) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/1)/(a/3)/(a/5)/(a/7)/(a/9)/.../(a/N)$$

131) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(a/2)/(a/4)/(a/6)/(a/8)/(a/10)/.../(a/N)$$

132) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(1-(a/2))+(3-(a/4))+(5-(a/6))+...+N$$

133) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S=(2-(a/1))+(4-(a/3))+(6-(a/5))+...+N$$

134) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S= (1-(a/2))^2 + (3-(a/4))^2 + (5-(a/6))^2 +...+ N$$

135) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S= (2-(a/1))^2 + (4-(a/3))^2 + (6-(a/5))^2 +...+ N$$

136) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S= (1-(a/2))^3 + (3-(a/4))^3 + (5-(a/6))^3 +...+ N$$

137) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:

$$S= (2-(a/1))^3 + (4-(a/3))^3 + (6-(a/5))^3 +...+ N$$

138) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
$$S = (1 - (a/2))^n + (3 - (a/4))^n + (5 - (a/6))^n + \dots + N$$

139) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
$$S = (2 - (a/1))^n + (4 - (a/3))^n + (6 - (a/5))^n + \dots + N$$

140) WRITE A PROGRAM TO PRINT
ANSWER OF THE FOLLOWING SERIES:
$$S = 1 + 2 + 5 + 10 + \dots + N$$

141) WRITE A PROGRAM TO PRINT THE
FOLLOWING SERIES:
$$S = 1, 2, 3, 4, 5, \dots, N$$

142) WRITE A PROGRAM TO PRINT THE
FOLLOWING SERIES:
$$S = 1, 3, 5, 7, 9, \dots, N$$

143) WRITE A PROGRAM TO PRINT THE
FOLLOWING SERIES:
$$S = 2, 4, 6, 8, 10, \dots, N$$

144) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,4,9,16,25,\dots,N$$

145) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,8,27,64,125,\dots,N$$

146) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=9,99,8,89,7,79,\dots,N$$

147) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,2,3,4,5,6,\dots,N$$

148) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,3,5,7,9,11,\dots,N$$

149) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,2,4,6,8,10,12,\dots,N$$

150) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,-3,5,-7,9,-11,\dots,N$$

151) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,-1,3,-5,7,-9,11,\dots,N$$

152) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,2,-4,6,-8,10,-12,\dots,N$$

153) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,-2,4,-6,8,-10,12,\dots,N$$

154) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,2,3,6,\dots,N$$

155) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,2,3,6,\dots,N$$

156) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,2,5,10,17,\dots,N$$

157) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,1,2,3,5,8,13,\dots,N$$

158) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,1,1,3,5,9,\dots,N$$

159) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,1,1,3,5,9,\dots,N$$

160) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,1,1,2,2,3,4,5,7,\dots,N$$

161) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,12,123,1234,12345,\dots,N$$

162) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=1,12,123,1234,12345,\dots,N$$

163) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,8,9,64,25,216,49,512,81,1000,\dots,N$$

164) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,4,27,16,125,36,343,64,729,100,\dots,N$$

165) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,4,27,16,125,36,343,64,729,100,\dots,N$$

166) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,1,2,4,3,5,7,6,8,10,12,9,11,13,15,17,\dots,N$$

167) WRITE A PROGRAM TO PRINT THE FOLLOWING SERIES:

$$S=0,2,1,3,4,6,8,5,7,9,11,10,12,14,16,18,\dots,N$$

=> 5. ARRAYS

1) WRITE A PROGRAM TO CREATE A SIMPLE DICTIONARY USING TWO ARRAYS (ONE FOR KEYS, ONE FOR VALUES).

2) WRITE A PROGRAM TO SHIFT ALL NEGATIVE NUMBERS TO THE BEGINNING AND POSITIVE NUMBERS TO THE END OF AN ARRAY.

3) WRITE A PROGRAM TO FIND THE SECOND LARGEST ELEMENT IN AN ARRAY WITHOUT SORTING.

4) WRITE A PROGRAM TO REARRANGE AN ARRAY SUCH THAT EVEN INDEX POSITIONS HOLD EVEN NUMBERS AND ODD INDEX POSITIONS HOLD ODD NUMBERS.

5) WRITE A PROGRAM TO CALCULATE THE CUMULATIVE SUM OF ELEMENTS IN AN ARRAY.

6) WRITE A PROGRAM TO FIND DUPLICATE ELEMENTS IN AN ARRAY.

7) WRITE A PROGRAM TO CHECK IF AN ARRAY IS A SUBARRAY OF ANOTHER ARRAY.

8) WRITE A PROGRAM TO FIND THE LARGEST AND SMALLEST ELEMENT IN AN ARRAY.

9) WRITE A PROGRAM TO REVERSE AN ARRAY.

10) WRITE A PROGRAM TO FIND THE FREQUENCY OF EACH ELEMENT IN AN ARRAY.

11) WRITE A PROGRAM TO ROTATE AN ARRAY TO THE RIGHT BY K STEPS.

12) WRITE A PROGRAM TO FIND THE LONGEST CONTINUOUS SUBARRAY THAT FORMS AN INCREASING SEQUENCE.

13) WRITE A PROGRAM TO REMOVE ALL NEGATIVE NUMBERS FROM AN ARRAY.

14) WRITE A PROGRAM TO CHECK IF ALL ELEMENTS IN AN ARRAY ARE DISTINCT.

15) WRITE A PROGRAM TO FIND THE MISSING ELEMENT FROM A PERMUTED ARRAY.

16) WRITE A PROGRAM TO FIND ALL UNIQUE ELEMENTS IN TWO ARRAYS (UNION OF TWO ARRAYS).

17) WRITE A PROGRAM TO CREATE A SNAKE PATTERN FROM AN ARRAY OF ELEMENTS.

18) WRITE A PROGRAM TO SORT AN ARRAY IN AN ALTERNATING ASCENDING AND DESCENDING ORDER.

19) WRITE A PROGRAM TO COMPUTE THE INNER PRODUCT OF TWO ARRAYS.

20) WRITE A PROGRAM TO REARRANGE ELEMENTS IN AN ARRAY IN REVERSE ZIG-ZAG ORDER.

21) WRITE A PROGRAM TO CHECK IF TWO ARRAYS ARE EQUAL OR NOT.

22) WRITE A PROGRAM TO FIND ALL PAIRS IN AN ARRAY WITH A GIVEN DIFFERENCE.

23) WRITE A PROGRAM TO ROTATE AN ARRAY SO THAT ALL PRIME NUMBERS COME FIRST.

24) WRITE A PROGRAM TO FIND THE FIRST DUPLICATE ELEMENT IN AN ARRAY.

25) WRITE A PROGRAM TO REORDER AN ARRAY TO PUT ODD NUMBERS BEFORE EVEN NUMBERS.

26) WRITE A PROGRAM TO REMOVE DUPLICATE ELEMENTS FROM AN ARRAY.

27) WRITE A PROGRAM TO MERGE TWO SORTED ARRAYS.

28) WRITE A PROGRAM TO FIND THE LONGEST SUBARRAY WITH ALTERNATING POSITIVE AND NEGATIVE NUMBERS.

29) WRITE A PROGRAM TO CALCULATE THE MODE (MOST FREQUENT ELEMENT) OF AN ARRAY.

30) WRITE A PROGRAM TO FIND THE PAIR OF ELEMENTS WITH THE MAXIMUM PRODUCT IN AN ARRAY.

31) WRITE A PROGRAM TO ROTATE ELEMENTS IN AN ARRAY CLOCKWISE BY HALF THE ARRAY LENGTH.

32) WRITE A PROGRAM TO COUNT THE NUMBER OF PAIRS IN AN ARRAY THAT HAVE A GIVEN SUM.

33) WRITE A PROGRAM TO CALCULATE THE DIFFERENCE BETWEEN SUM OF SQUARES OF EVEN AND ODD ELEMENTS.

34) WRITE A PROGRAM TO REARRANGE AN ARRAY IN THE ORDER OF ITS ELEMENT FREQUENCIES.

35) WRITE A PROGRAM TO CHECK IF AN ARRAY IS SYMMETRIC AROUND ITS MIDDLE.

36) WRITE A PROGRAM TO REMOVE ELEMENTS THAT ARE MULTIPLES OF A GIVEN NUMBER FROM AN ARRAY.

37) WRITE A PROGRAM TO PARTITION AN ARRAY INTO TWO GROUPS OF EQUAL SUM.

38) WRITE A PROGRAM TO CHECK IF ELEMENTS OF AN ARRAY CAN FORM AN ARITHMETIC SEQUENCE.

39) WRITE A PROGRAM TO FIND THE SUM OF THE LONGEST CONTINUOUS SUBARRAY WITH ONLY POSITIVE NUMBERS.

40) WRITE A PROGRAM TO CALCULATE THE SUM OF DIFFERENCES BETWEEN ALL POSSIBLE PAIRS OF AN ARRAY.

41) WRITE A PROGRAM TO SORT AN ARRAY OF FLOATING-POINT NUMBERS.

42) WRITE A PROGRAM TO FIND THE MAXIMUM GAP BETWEEN CONSECUTIVE ELEMENTS IN A SORTED ARRAY.

43) WRITE A PROGRAM TO CONVERT AN ARRAY INTO A MAX HEAP STRUCTURE.

44) WRITE A PROGRAM TO FIND THE RANGE (DIFFERENCE BETWEEN MAX AND MIN) OF ALL SUBARRAYS OF LENGTH K.

45) WRITE A PROGRAM TO FIND THE SMALLEST ELEMENT GREATER THAN A GIVEN VALUE IN A SORTED ARRAY.

46) WRITE A PROGRAM TO REORDER AN ARRAY SO THAT IT STARTS AND ENDS WITH ITS MINIMUM ELEMENT.

47) WRITE A PROGRAM TO COUNT THE NUMBER OF SUBARRAYS WITH A GIVEN SUM.

48) WRITE A PROGRAM TO FIND THE NUMBER OF TRIPLETS IN AN ARRAY WITH A GIVEN SUM.

49) WRITE A PROGRAM TO REVERSE THE ORDER OF ELEMENTS IN A SPECIFIC RANGE OF AN ARRAY.

50) WRITE A PROGRAM TO CREATE AN ARRAY THAT CONTAINS THE RUNNING AVERAGE OF ELEMENTS.

51) WRITE A PROGRAM TO FIND ALL SUBARRAYS OF AN ARRAY THAT HAVE AN EVEN NUMBER OF ELEMENTS.

52) WRITE A PROGRAM TO FIND THE COMMON ELEMENTS IN THREE SORTED ARRAYS.

53) WRITE A PROGRAM TO ROTATE AN ARRAY BY 180 DEGREES.

54) WRITE A PROGRAM TO DISPLAY THE SUM OF ELEMENTS AT PRIME INDEX POSITIONS IN AN ARRAY.

55) WRITE A PROGRAM TO FIND ALL DISTINCT PERMUTATIONS OF AN ARRAY WITH DUPLICATES.

56) WRITE A PROGRAM TO MERGE K SORTED ARRAYS INTO A SINGLE SORTED ARRAY.

57) WRITE A PROGRAM TO ROTATE ELEMENTS IN AN ARRAY COUNTERCLOCKWISE BY HALF THE ARRAY LENGTH.

58) WRITE A PROGRAM TO FIND THE MINIMUM PRODUCT OF TWO ELEMENTS IN AN ARRAY.

59) WRITE A PROGRAM TO FIND THE PAIR WITH THE LARGEST SUM IN AN UNSORTED ARRAY.

60) WRITE A PROGRAM TO FIND THE FIRST AND LAST OCCURRENCES OF A GIVEN ELEMENT IN A SORTED ARRAY.

61) WRITE A PROGRAM TO REPLACE ALL NEGATIVE VALUES WITH ZERO IN AN ARRAY.

62) WRITE A PROGRAM TO CREATE AN ARRAY WITH ELEMENTS THAT ARE EITHER GREATER THAN OR EQUAL TO THEIR NEIGHBORS.

63) WRITE A PROGRAM TO SPLIT AN ARRAY OF INTEGERS INTO TWO GROUPS WITH THE SAME SUM.

64) WRITE A PROGRAM TO COUNT ALL SUBARRAYS WHERE SUM OF ELEMENTS IS A PERFECT SQUARE.

65) WRITE A PROGRAM TO FIND THE LONGEST PALINDROMIC SUBARRAY IN A GIVEN ARRAY.

66) WRITE A PROGRAM TO FIND THE MINIMUM DIFFERENCE BETWEEN TWO ELEMENTS IN AN ARRAY.

67) WRITE A PROGRAM TO FIND ALL ELEMENTS THAT APPEAR MORE THAN $N/3$ TIMES IN AN ARRAY.

68) WRITE A PROGRAM TO FIND THE SHORTEST SUBARRAY WITH THE MAXIMUM SUM.

69) WRITE A PROGRAM TO FIND THE MEDIAN OF AN ARRAY WITHOUT SORTING.

70) WRITE A PROGRAM TO REMOVE ALL ZEROS AND SHIFT NON-ZERO ELEMENTS TO THE LEFT IN AN ARRAY.

71) WRITE A PROGRAM TO FIND THE FREQUENCY OF EACH PAIR SUM IN AN ARRAY.

72) WRITE A PROGRAM TO REPLACE EVERY ELEMENT WITH THE PRODUCT OF OTHER ELEMENTS IN AN ARRAY.

73) WRITE A PROGRAM TO FIND THE LONGEST SUBARRAY WITH SUM LESS THAN OR EQUAL TO K.

74) WRITE A PROGRAM TO CREATE A 1D ARRAY FROM A GIVEN 2D ARRAY BY FLATTENING.

75) WRITE A PROGRAM TO CHECK IF ELEMENTS OF AN ARRAY FORM A FIBONACCI SEQUENCE.

76) WRITE A PROGRAM TO ROTATE THE ARRAY LEFT BY A GIVEN NUMBER OF POSITIONS.

77) WRITE A PROGRAM TO FIND THE DIFFERENCE BETWEEN MAXIMUM AND MINIMUM IN EACH SUBARRAY.

78) WRITE A PROGRAM TO COUNT ELEMENTS THAT ARE GREATER THAN ALL PRECEDING ELEMENTS.

79) WRITE A PROGRAM TO REORDER AN ARRAY BY MOVING ALL ZEROES TO THE END.

80) WRITE A PROGRAM TO FIND THE SMALLEST ELEMENT GREATER THAN THE MAXIMUM OF ANOTHER ARRAY.

81) WRITE A PROGRAM TO CONVERT AN ARRAY INTO A MIN HEAP STRUCTURE.

82) WRITE A PROGRAM TO FIND THE PRODUCT OF ALL ELEMENTS IN AN ARRAY EXCEPT ITSELF.

83) WRITE A PROGRAM TO FIND THE K SMALLEST ELEMENTS IN AN UNSORTED ARRAY.

84) WRITE A PROGRAM TO CALCULATE THE AVERAGE OF ALL SUBARRAYS OF SIZE K.

85) WRITE A PROGRAM TO FIND THE FIRST MISSING POSITIVE INTEGER IN AN ARRAY.

86) WRITE A PROGRAM TO COMPUTE THE CROSS SUM OF ALL PAIRS IN AN ARRAY.

87) WRITE A PROGRAM TO FIND THE NUMBER OF ELEMENTS THAT CAN FORM A PAIR SUM TO A PRIME NUMBER.

88) WRITE A PROGRAM TO ROTATE THE FIRST HALF OF AN ARRAY BY A GIVEN NUMBER OF POSITIONS.

89) WRITE A PROGRAM TO FIND THE COUNT OF ELEMENTS GREATER THAN A GIVEN VALUE.

90) WRITE A PROGRAM TO FIND THE MEAN, MEDIAN, AND MODE OF AN ARRAY.

91) WRITE A PROGRAM TO FIND THE K MOST FREQUENT ELEMENTS IN AN ARRAY.

92) WRITE A PROGRAM TO CALCULATE THE RANGE OF ELEMENTS IN ALL SUBARRAYS OF LENGTH K.

93) WRITE A PROGRAM TO FIND THE LARGEST SUBARRAY THAT FORMS A CONSECUTIVE SEQUENCE.

94) WRITE A PROGRAM TO FIND THE K LARGEST ELEMENTS IN AN ARRAY.

95) WRITE A PROGRAM TO FIND THE MAXIMUM SUBARRAY SUM UNDER A GIVEN BOUND.

96) WRITE A PROGRAM TO CALCULATE THE NUMBER OF INVERSIONS IN AN ARRAY.

97) WRITE A PROGRAM TO CHECK IF AN ARRAY IS MONOTONIC.

98) WRITE A PROGRAM TO FIND ALL ELEMENTS THAT ARE NOT FOLLOWED BY LARGER ELEMENTS.

99) WRITE A PROGRAM TO SORT AN ARRAY IN WAVE FORM.

100) WRITE A PROGRAM TO COMPUTE THE CUMULATIVE PRODUCT OF ELEMENTS IN AN ARRAY.

101) WRITE A PROGRAM TO FIND THE SMALLEST SUBARRAY WITH A GIVEN SUM.

102) WRITE A PROGRAM TO FIND THE LONGEST SUBARRAY WITH AT LEAST K DISTINCT ELEMENTS.

103) WRITE A PROGRAM TO REVERSE AN ARRAY IN PLACE USING A SINGLE LOOP.

104) WRITE A PROGRAM TO FIND THE KTH LARGEST ELEMENT IN AN UNSORTED ARRAY USING QUICKSELECT.

105) WRITE A PROGRAM TO MERGE TWO SORTED LINKED LISTS INTO A SINGLE SORTED ARRAY.

106) WRITE A PROGRAM TO PARTITION AN ARRAY AROUND A PIVOT ELEMENT.

107) WRITE A PROGRAM TO FIND ALL PAIRED ELEMENTS THAT HAVE A PRODUCT GREATER THAN A GIVEN NUMBER.

108) WRITE A PROGRAM TO GENERATE ALL POSSIBLE SUBARRAYS OF AN ARRAY.

109) WRITE A PROGRAM TO COUNT HOW MANY TIMES EACH ELEMENT APPEARS IN AN ARRAY USING A HASHMAP.

110) WRITE A PROGRAM TO FIND THE GCD (GREATEST COMMON DIVISOR) OF ALL ELEMENTS IN AN ARRAY.

111) WRITE A PROGRAM TO CREATE A NEW ARRAY WITH THE SQUARE OF EACH ELEMENT IN THE ORIGINAL ARRAY.

112) WRITE A PROGRAM TO SHIFT AN ARRAY LEFT BY ONE POSITION.

113) WRITE A PROGRAM TO FIND THE SUM OF ALL ODD INDEXED ELEMENTS IN AN ARRAY.

114) WRITE A PROGRAM TO REARRANGE AN ARRAY IN SUCH A WAY THAT THE ELEMENTS ARE IN A ZIGZAG PATTERN.

115) WRITE A PROGRAM TO CREATE A CIRCULAR ARRAY.

116) WRITE A PROGRAM TO FIND THE MOST FREQUENT ELEMENT IN AN ARRAY.

117) WRITE A PROGRAM TO CLONE AN ARRAY OF OBJECTS.

118) WRITE A PROGRAM TO REMOVE ELEMENTS THAT OCCUR MORE THAN A SPECIFIED NUMBER OF TIMES.

119) WRITE A PROGRAM TO FIND THE ELEMENT WITH THE MINIMUM ABSOLUTE DIFFERENCE TO A TARGET VALUE.

120) WRITE A PROGRAM TO COUNT THE NUMBER OF SUBARRAYS WITH AN EVEN SUM.

=> 6. PATTERNS

1. STAR PATTERNS

2. NUMBER PATTERNS

3. ALPHABET PATTERNS

1. STAR PATTERNS

(1) * * * * *

* * * * *

* * * * *

* * * * *

* * * * *

(2) *

* *

* * *

* * * *

* * * * *

(3) * * * * *

* * * *

* * *

* *

*

(4) *

* *

* * *

* * * *

* * * * *

(5) * * * * *

* * * *

* * *

* *

*

(6) *

* *

* * *

* * * *

* * * * *

(7) * * * * *

* * * *

* * *

* *

*

(8) *

* *

* * *

* * * *

* * *

* *

*

(9) *

* *

* * *

* * * *

* * *

* *

*

(10) * * * * *

* * * * *

* * * * *

* * * * *

* * * * *

(11) * * * * *

* * * * *

* * * * *

* * * * *

* * * * *

(12) * * * * *

* *

* *

* *

* * * * *

(13)

*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

(14)

* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *

(15)

* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *

(16)

* * * * * * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

(17)

```
*
*  *
*    *
*      *
*        *
*          *
*            *
*              *
*                *
```

(18)

```
*
*      *
*        *
*          *
*            *
*              *
*                *
*                  *
*                    *
```

(19)

```
      *
    *  *
  *    *
*      *
*        *
*          *
*            *
*              *
*                *
*                  *
*                    *
```

(20)

```
* * * * *
*           *
*             *
*               *
*                 *
*                   *
*                     *
*                       *
```

<p>(21)</p> <pre> * * * * * * * * * * * * * * * * </pre>	<p>(22)</p> <pre> * * * * * * * * * * * * * * * * </pre>
--	--

<p>(23)</p> <pre> * * * * * * * * * * * * * * </pre>	<p>(24)</p> <pre> * </pre>
--	--

(25)

```

*
*  *
*    *
*      *
*        *
* * * * *

```

(26)

```

* * * * *
*       *
*       *
*       *
*       *
*       *

```

(27)

```

      *
    *  *
  *    *
*      *
*      *
* * * * *

```

(28)

```

* * * * *
*       *
*       *
*       *
*       *
*

```

(29)

```

* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *

```

(30)

```

* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *

```


(31)

```
* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *
```

(32)

```
* * * * *
*      *
*      *
* *
*
* *
*      *
*      *
* * * * *
```

(33)

```
*          *
* *          * *
*      *      *      *
*          *          *
*      *      *      *
* *          * *
*          *
```

(34)

```
* * * * * * * * * *
* * * *      * * * *
* * *      * * *
* *          * *
*          *
* *          * *
* * *      * * *
* * * *      * * * *
* * * * * * * * * *
```

(35)

```
* * * * * * * * * * * *
*           *           *
*           *           *
*           *           *
*           *           *
*           *           *
* * * * * * * * * * * *
*           *           *
*           *           *
*           *           *
*           *           *
*           *           *
* * * * * * * * * * * *
```

(36)

```
* * * * * * * * * * * * * * * * * * * * * * * * * *
*                                           *
*                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * *
*                                           *
*                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * *
```

(37)

```
*               * * * * * * * *
*               *
*               *
*               *
*               *
* * * * * * * * * * * * * * * *
*               *               *
*               *               *
*               *               *
* * * * * * * * * *               *
```

(38)

```
*           *
*         *
*       *
*     *
*   *
* *
*   *
*     *
*       *
```

(39)

```
      *
    *  *
  *      *
*        *
* * * * *
*        *
  *      *
    *  *
      *
```

2. NUMBER PATTERNS

(1)	11111	(2)	1111111111
	11111		1111111111
	11111		1111111111
	11111		
	11111		

(3)	1	(4)	1 1 1 1 1	(5)	1
	1 1		1 1 1 1		1 1
	1 1 1		1 1 1		1 1 1
	1 1 1 1		1 1		1 1 1 1
	1 1 1 1 1		1		1 1 1 1 1

(6)	1 1 1 1 1	(7)	1	(8)	1 1 1 1 1
	1 1 1 1		1 1		1 1 1 1
	1 1 1		1 1 1		1 1 1
	1 1		1 1 1 1		1 1
	1		1 1 1 1 1		1

(9)	1	(10)	1
	1 1		1 1
	1 1 1		1 1 1
	1 1 1 1		1 1 1 1
	1 1 1		1 1 1
	1 1		1 1
	1		1

(11)	1 1 1 1 1	(12)	1 1 1 1 1
	1 1 1 1 1		1 1 1 1 1
	1 1 1 1 1		1 1 1 1 1
	1 1 1 1 1		1 1 1 1 1
	1 1 1 1 1		1 1 1 1 1

(13)	1	(14)	1	(15)	1
	1 1		1 1		1 1
	1 1 1		1 1 1		1 1 1
	1 1 1 1		1 1 1 1		1 1 1 1
	1 1 1 1 1		1 1 1 1 1		1 1 1 1 1
	1 1 1 1		1 1 1 1		1 1 1 1
	1 1 1		1 1 1		1 1 1
	1 1		1 1		1 1
	1		1		1

(16) 1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

(17) 1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

(18) 1

1 1

1 1 1

1 1 1 1

1 1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

(19) 1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

(20) 1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

(21)

1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

1 1

1 1 1

1 1 1 1

1 1 1 1 1

(22) 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1

1 1 1 1 1

1 1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

(1) 1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

5 5 5 5 5

(2) 1 1 1 1 1 1 1 1 1 1

2 2 2 2 2 2 2 2 2 2

3 3 3 3 3 3 3 3 3 3

(3) 1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

(4) 1 1 1 1 1

2 2 2 2

3 3 3

2 2

5

(5) 1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

(6) 1 1 1 1 1

2 2 2 2

3 3 3

4 4

5

(7) 1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

(8) 1 1 1 1 1

2 2 2 2

3 3 3

4 4

5

(9)	1	(10)	1
	2 2		2 2
	3 3 3		3 3 3
	4 4 4 4		4 4 4 4
	5 5 5		5 5 5
	6 6		6 6
	7		7

(11)	1 1 1 1 1	(12)	1 1 1 1 1
	2 2 2 2 2		2 2 2 2 2
	3 3 3 3 3		3 3 3 3 3
	4 4 4 4 4		4 4 4 4 4
	5 5 5 5 5		5 5 5 5 5

(13)	1	(14)	1	(15)	1
	2 2		2 2		2 2
	3 3 3		3 3 3		3 3 3
	4 4 4 4		4 4 4 4		4 4 4 4
	5 5 5 5 5		5 5 5 5 5		5 5 5 5 5
	6 6 6 6		6 6 6 6		6 6 6 6
	7 7 7		7 7 7		7 7 7
	8 8		8 8		8 8
	9		9		9

(16)	1 1 1 1 1	(17)	1 1 1 1 1
	2 2 2 2		2 2 2 2
	3 3 3		3 3 3
	4 4		4 4
	5		5
	6 6		6 6
	7 7 7		7 7 7
	8 8 8 8		8 8 8 8
	9 9 9 9 9		9 9 9 9 9

(18) 1	1	(19) 1 1 1 1 1
2 2	2 2	2 2 2 2
3 3 3	3 3 3	3 3 3
4 4 4 4	4 4 4 4	4 4
5 5 5 5 5 5 5 5 5 5		5
6 6 6 6	6 6 6 6	6 6
7 7 7	7 7 7	7 7 7
8 8	8 8	8 8 8 8
9	9	9 9 9 9 9

(20) 1 1 1 1 1	(21) 1 1 1 1 1
2 2 2 2	2 2 2 2
3 3 3	3 3 3
4 4	4 4
5	5
6 6	6 6
7 7 7	7 7 7
8 8 8 8	8 8 8 8
9 9 9 9 9	9 9 9 9 9

(22) 1 1 1 1 1 1 1 1 1 1

2 2 2 2 2 2 2 2

3 3 3 3 3 3

4 4 4 4

5 5

6 6 6 6

7 7 7 7 7 7

8 8 8 8 8 8 8 8

9 9 9 9 9 9 9 9 9 9

(1) 1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

(2) 1 2 3 4 5 6 7

8 9 10 11 12 13 14 15

16 17 18 19 20 21 22

(3) 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

(4) 1 2 3 4 5

6 7 8 9

10 11 12

13 14

15

(5)

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

(6) 1 2 3 4 5	(7) 1	(8) 1 2 3 4 5
6 7 8 9	2 3	6 7 8 9
10 11 12	4 5 6	10 11 12
13 14	7 8 9 10	13 14
15	11 12 13 14 15	15

(1) 1 2 3 4 5	(2) 1 2 3 4 5 6 7 8 9
1 2 3 4 5	1 2 3 4 5 6 7 8 9
1 2 3 4 5	1 2 3 4 5 6 7 8 9
1 2 3 4 5	
1 2 3 4 5	

(3) 1	(4) 1 2 3 4 5	(5) 5
1 2	1 2 3 4	4 5
1 2 3	1 2 3	3 4 5
1 2 3 4	1 2	2 3 4 5
1 2 3 4 5	1	1 2 3 4 5

(6) 1 2 3 4 5

2 3 4 5

3 4 5

4 5

5

(7) 5

4 5 6

3 4 5 6 7

2 3 4 5 6 7 8

1 2 3 4 5 6 7 8 9

(8) 1 2 3 4 5 6 7 8 9

2 3 4 5 6 7 8

3 4 5 6 7

4 5 6

5

(9) 1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

1 2 3 4 3 2 1

1 2 3 2 1

1 2 1

1

(10) 5

5 4 5

5 4 3 4 5

5 4 3 2 3 4 5

5 4 3 2 1 2 3 4 5

5 4 3 2 3 4 5

5 4 3 4 5

5 4 5

5

(11) 1	(12) 1	(13) 5 4 3 2 1
2 1	1 2	4 3 2 1
3 2 1	1 2 3	3 2 1
4 3 2 1	1 2 3 4	2 1
5 4 3 2 1	1 2 3 4 5	1

(14) 1 2 3 4 5	(15) 5	(16) 5
1 2 3 4	4 5	5 4
1 2 3	3 4 5	5 4 3
1 2	2 3 4 5	5 4 3 2
1	1 2 3 4 5	5 4 3 2 1

(17) 1 2 3 4 5	(18) 5 4 3 2 1
2 3 4 5	5 4 3 2
3 4 5	5 4 3
4 5	5 4
5	5

(19) 1

2 3 2

4 5 6 5 4

7 8 9 0 9 8 7

(20) 0

0 1

0 2 4

0 3 6 9

0 4 8 12 16

(21) 1

0 1

1 0 1

0 1 0 1

1 0 1 0 1

(22) 0

1 0

0 1 0

1 0 1 0

0 1 0 1 0

(23) 1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

(24) 2

3 5

7 11 13

17 19 23 29

31 37 41 43 47

(25)

1
1 2 1
1 2 4 2 1
1 2 4 8 4 2 1
1 2 4 8 16 8 4 2 1
1 2 4 8 16 32 16 8 4 2 1
1 2 4 8 16 32 64 32 16 8 4 2 1

(26)	1	(27)	1
	2 3 2		0 1
	3 4 5 4 3		0 1 0
	4 5 6 7 6 5 4		1 0 1 0
	5 6 7 8 9 8 7 6 5		1 0 1 0 1

(28)	1	(29)	1 0 1 0 1 0 1	(30)	15
	1 0 1		1 0 1 0 1		14 10
	1 0 1 0 1		1 0 1		13 9 6
	1 0 1 0 1 0 1		1		12 8 5 3
					11 7 4 2 1

(31)	15	(32)	1 2 4 7 11	(33)	11 7 4 2 1
	10 14		3 5 8 12		12 8 5 3
	6 9 13		6 9 13		13 9 6
	3 5 8 12		10 14		14 10
	1 2 4 7 11		15		15

(34)	1 2 3 4 5	(35)	5 4 3 2 1
	10 9 8 7 6		6 7 8 9 10
	11 12 13 14 15		15 14 13 12 11
	20 19 18 17 16		16 17 18 19 20

(36)	20 19 18 17 16	(37)	16 17 18 19 20
	11 12 13 14 15		15 14 13 12 11
	10 9 8 7 6		6 7 8 9 10
	1 2 3 4 5		5 4 3 2 1

(38)	1 10 11 20	(39)	5 6 15 16
	2 9 12 19		4 7 14 17
	3 8 13 18		3 8 13 18
	4 7 14 17		2 9 12 19
	5 6 15 16		1 10 11 20

(40) 1 5 1 5 1	(41) 1 2 3 4 5
2 4 2 4 2	5 4 3 2 1
3 3 3 3 3	1 2 3 4 5
4 2 4 2 4	5 4 3 2 1
5 1 5 1 5	1 2 3 4 5

(42) 1	(43) 2	(44) 1
2 4	1 3	3 5
1 3 5	2 4 6	7 9 11
2 4 6 8	1 3 5 7	13 15 17 19
1 3 5 7 9	2 4 6 8 10	21 23 25 27 29

(45) 2	(46) 1 2 3 4 5
4 6	16 17 18 19 6
8 10 12	15 24 25 20 7
14 16 18 20	14 23 22 21 8
22 24 26 28 30	13 12 11 10 9

(47) 5 4 3 2 1	(48) 13 12 11 10 9
6 19 18 17 16	14 23 22 21 8
7 20 25 24 15	15 24 24 20 7
8 21 22 23 14	16 17 18 19 6
9 10 11 12 13	1 2 3 4 5

(49) 9 10 11 12 13	(50) 1 2 3 4 5
8 21 22 23 14	6 7 8 9 10
7 20 25 24 15	1 2 3 4 5
6 19 18 17 16	6 7 8 9 10
5 4 3 2 1	1 2 3 4 5

(51) 6 7 8 9 10	(52) 1 2 3 4 5	(53) 10 9 8 7 6
1 2 3 4 5	10 9 8 7 6	1 2 3 4 5
6 7 8 9 10	1 2 3 4 5	10 9 8 7 6
1 2 3 4 5	10 9 8 7 6	1 2 3 4 5
6 7 8 9 10	1 2 3 4 5	10 9 8 7 6

(54)	1	6	1	6	(55)	6	1	6	1	(56)	1	10	1	10
	2	7	2	7		7	2	7	2		2	9	2	9
	3	8	3	8		8	3	8	3		3	8	3	8
	4	9	4	9		9	4	9	4		4	7	4	7
	5	10	5	10		10	5	10	5		5	6	5	6

(57)	10	1	10	1	(58)	5	5	5	5	5	5	5	5
	9	2	9	2		5	4	4	4	4	4	4	5
	8	3	8	3		5	4	3	3	3	3	4	5
	7	4	7	4		5	4	3	2	2	2	3	4
	6	5	6	5		5	4	3	2	1	2	3	4
						5	4	3	2	2	2	3	4
						5	4	3	3	3	3	4	5
						5	4	4	4	4	4	4	5
						5	5	5	5	5	5	5	5

(59) 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5
5 4 3 3 3 3 3 3 3 3 3 3 3 3 4 5
5 4 3 2 2 2 2 2 2 2 2 2 2 3 4 5
5 4 3 2 1 1 1 1 1 1 1 1 2 3 4 5
5 4 3 2 2 2 2 2 2 2 2 2 2 3 4 5
5 4 3 3 3 3 3 3 3 3 3 3 3 3 4 5
5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

(60) 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1

(61) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1
 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1
 1 2 3 4 5 5 5 5 5 5 5 5 5 5 4 3 2 1
 1 2 3 4 4 4 4 4 4 4 4 4 4 4 3 2 1
 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1
 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

(62) 0 0 0 0 0 0 0 0 0	(63) 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 0	1 0 0 0 0 0 0 0 1
0 1 1 1 1 1 1 1 0	1 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1

(64) 0 0 0 0	(65) 1 1 1 1
0 1 1 0	1 0 0 1
0 1 1 0	1 0 0 1
0 1 1 0	1 0 0 1
0 1 1 0	1 0 0 1
0 1 1 0	1 0 0 1
0 1 1 0	1 0 0 1
0 1 1 0	1 0 0 1
0 0 0 0	1 1 1 1

3. ALPHABET PATTERNS

(1) a a a a a	(2) a a a a a a a a a a
a a a a a	a a a a a a a a a a
a a a a a	a a a a a a a a a a
a a a a a	
a a a a a	

(3) a	(4) a a a a a	(5) a
a a	a a a a	a a
a a a	a a a	a a a
a a a a	a a	a a a a
a a a a a	a	a a a a a

(6) a a a a a	(7) a	(8) a a a a a
a a a a	a a	a a a a
a a a	a a a	a a a
a a	a a a a	a a
a	a a a a a	a

(9) a	(10) a
a a	a a
a a a	a a a
a a a a	a a a a
a a a	a a a
a a	a a
a	a

(11) a a a a a
a a a a a
a a a a a
a a a a a
a a a a a

(12) a a a a a
a a a a a
a a a a a
a a a a a
a a a a a

(1) a a a a a
b b b b b
c c c c c
d d d d d
e e e e e

(2) a a a a a a a a a a
b b b b b b b b b b
c c c c c c c c c c

(3) a
b b
c c c
d d d d
e e e e e

(4) a a a a a
b b b b
c c c
d d
e

(5) a
b b
c c c
d d d d
e e e e e

(6)	a a a a a	(7)	a	(8)	a a a a a
	b b b b		b b		b b b b
	c c c		c c c		c c c
	d d		d d d d		d d
	e		e e e e e		e

(9)	a	(10)	a
	b b		b b
	c c c		c c c
	d d d d		d d d d
	e e e		e e e
	f f		f f
	g		g

(11)	a a a a a	(12)	a a a a a
	b b b b b		b b b b b
	c c c c c		c c c c c
	d d d d d		d d d d d
	e e e e e		e e e e e

(13)	a	(14)	a	(15)	a
	b b		b b		b b
	c c c		c c c		c c c
	d d d d		d d d d		d d d d
	e e e e e		e e e e e		e e e e e
	f f f f		f f f f		f f f f
	g g g		g g g		g g g
	h h		h h		h h
	i		i		i

(16)	a a a a a	(17)	a a a a a
	b b b b		b b b b
	c c c		c c c
	d d		d d
	e		e
	f f		f f
	g g g		g g g
	h h h h		h h h h
	i i i i i		i i i i i

(18) a

b b

c c c

d d d d

e e e e e e e e e e

f f f f

g g g

h h

i

a

b b

c c c

d d d d

f f f f

g g g

h h

i

(19) a a a a a

b b b b

c c c

d d

e

f f

g g g

h h h h

i i i i i

(20) a a a a a

b b b b

c c c

d d

e

f f

g g g

h h h h

i i i i i

(21)

a a a a a

b b b b

c c c

d d

e

f f

g g g

h h h h

i i i i i

(22) a a a a a a a a a a

b b b b b b b b

c c c c c c

d d d d

e e

f f f f

g g g g g g

h h h h h h h h

i i i i i i i i i i

(1) a b c d e

a b c d e

a b c d e

a b c d e

a b c d e

(2) a b c d e f g h I j

a b c d e f g h I j

a b c d e f g h I j

(3)	a	(4)	a b c d e	(5)	a
	a b		a b c d		a b
	a b c		a b c		a b c
	a b c d		a b		a b c d
	a b c d e		a		a b c d e

(6)	a b c d e	(7)	a	(8)	a b c d e
	a b c d		a b		a b c d
	a b c		a b c		a b c
	a b		a b c d		a b
	a		a b c d e		a

(9)	a	(10)	a
	a b		a b
	a b c		a b c
	a b c d		a b c d
	a b c		a b c
	a b		a b
	a		a

(11) a b c d e
 a b c d e
 a b c d e
 a b c d e
 a b c d e

(12) a b c d e
 a b c d e
 a b c d e
 a b c d e
 a b c d e

=> 7. STRINGS

1) WRITE A PROGRAM TO CHECK IF A CHARACTER IS IN THE RANGE OF A-Z.

2) WRITE A PROGRAM TO CHECK IF A CHARACTER IS A SPECIAL CHARACTER.

3) WRITE A PROGRAM TO DISPLAY A CHARACTER BASED ON ITS ASCII VALUE (e.g., 65 = A).

4) WRITE A PROGRAM TO DISPLAY A ASCII VALUE BASED ON CHARACTER (e.g., A = 65).

5) WRITE A PROGRAM TO CHECK IF A CHARACTER IS AN ALPHABET.

6) WRITE A PROGRAM TO CHECK IF A CHARACTER IS A DIGIT.

7) WRITE A PROGRAM TO CHECK IF A CHARACTER IS VOWEL OR CONSONANT.

8) WRITE A PROGRAM TO DISPLAY THE NTH CHARACTER OF A STRING.

9) WRITE A PROGRAM TO REMOVE WHITESPACE FROM A STRING.

10) WRITE A PROGRAM TO FIND THE LENGTH OF A STRING WITHOUT USING BUILT-IN FUNCTIONS.

11) WRITE A PROGRAM TO REPEAT A STRING N TIMES.

12) WRITE A PROGRAM TO CONVERT A STRING TO UPPERCASE AND LOWERCASE.

13) WRITE A PROGRAM TO CONCATENATE TWO STRINGS.

14) WRITE A PROGRAM TO COMPARE TWO STRINGS (CASE SENSITIVE AND INSENSITIVE).

15) WRITE A PROGRAM TO REPLACE ALL OCCURRENCES OF A CHARACTER IN A STRING.

16) WRITE A PROGRAM TO DISPLAY ALL UNIQUE CHARACTERS FROM A STRING.

17) WRITE A PROGRAM TO REMOVE VOWELS FROM A STRING.

18) WRITE A PROGRAM TO REMOVE CONSONANTS FROM A STRING.

19) WRITE A PROGRAM TO REMOVE DIGITS FROM A STRING.

20) WRITE A PROGRAM TO CHECK IF A STRING HAS ALL UNIQUE CHARACTERS.

21) WRITE A PROGRAM TO CHECK IF A STRING CONTAINS ONLY DIGITS.

22) WRITE A PROGRAM TO DETERMINE IF A STRING IS AN ANAGRAM OF ANOTHER STRING.

23) WRITE A PROGRAM TO CHECK IF A STRING IS A SUBSTRING OF ANOTHER STRING.

24) WRITE A PROGRAM TO FIND OUT IF A STRING IS A ROTATIONAL STRING OF ANOTHER STRING.

25) WRITE A PROGRAM TO CHECK IF A STRING IS A VALID EMAIL ADDRESS.

26) WRITE A PROGRAM TO CHECK IF A STRING CONTAINS A NUMBER.

27) WRITE A PROGRAM TO CHECK IF A STRING IS A VALID URL.

28) WRITE A PROGRAM TO CHECK IF A STRING IS A VALID PHONE NUMBER.

29) WRITE A PROGRAM TO CHECK IF A STRING ENDS WITH A SPECIFIC CHARACTER OR NOT.

30) WRITE A PROGRAM TO CHECK IF A STRING HAS SPACES OR NOT.

31) WRITE A PROGRAM TO REVERSE A STRING WITHOUT USING BUILT-IN FUNCTIONS.

32) WRITE A PROGRAM TO COUNT THE NUMBER OF VOWELS AND CONSONANTS IN A STRING.

33) WRITE A PROGRAM TO CHECK IF A STRING IS A PALINDROME.

34) WRITE A PROGRAM TO FIND THE FREQUENCY OF EACH CHARACTER IN A STRING.

35) WRITE A PROGRAM TO CONVERT THE FIRST LETTER OF EACH WORD IN A STRING TO UPPERCASE.

36) WRITE A PROGRAM TO FIND THE LONGEST WORD IN A STRING.

37) WRITE A PROGRAM TO FIND THE SHORTEST WORD IN A STRING.

38) WRITE A PROGRAM TO REMOVE DUPLICATE CHARACTERS FROM A STRING.

39) WRITE A PROGRAM TO DISPLAY THE FIRST NON-REPEATING CHARACTER IN A STRING.

40) WRITE A PROGRAM TO COUNT THE NUMBER OF WORDS IN A STRING.

41) WRITE A PROGRAM TO DISPLAY ALL PERMUTATIONS OF A STRING.

42) WRITE A PROGRAM TO CONVERT A STRING INTO ITS ASCII EQUIVALENT REPRESENTATION.

43) WRITE A PROGRAM TO SORT THE CHARACTERS IN A STRING ALPHABETICALLY.

44) WRITE A PROGRAM TO FIND THE MOST FREQUENT CHARACTER IN A STRING.

45) WRITE A PROGRAM TO CHECK IF TWO STRINGS ARE MIRROR IMAGES OF EACH OTHER.

46) WRITE A PROGRAM TO INSERT A SUBSTRING INTO A STRING AT A SPECIFIED POSITION.

47) WRITE A PROGRAM TO REMOVE ALL PUNCTUATION MARKS FROM A STRING.

48) WRITE A PROGRAM TO REVERSE EACH WORD IN A STRING WHILE KEEPING THE WORD ORDER INTACT.

49) WRITE A PROGRAM TO CHECK IF TWO STRINGS HAVE THE SAME SET OF CHARACTERS.

50) WRITE A PROGRAM TO CONVERT A STRING OF DIGITS INTO AN INTEGER.

51) WRITE A PROGRAM TO CAPITALIZE ONLY THE FIRST LETTER OF A STRING.

52) WRITE A PROGRAM TO REMOVE ALL CHARACTERS EXCEPT ALPHABETS FROM A STRING.

53) WRITE A PROGRAM TO REMOVE ALL CHARACTERS EXCEPT DIGITS FROM A STRING.

54) WRITE A PROGRAM TO REMOVE ALL SPECIAL CHARACTERS FROM A STRING.

55) WRITE A PROGRAM TO FIND THE FIRST OCCURRENCE OF A CHARACTER IN A STRING.

56) WRITE A PROGRAM TO FIND THE LAST OCCURRENCE OF A CHARACTER IN A STRING.

57) WRITE A PROGRAM TO COUNT THE OCCURRENCES OF A SUBSTRING IN A STRING.

58) WRITE A PROGRAM TO CONVERT A SENTENCE TO "TITLE CASE" (CAPITALIZE THE FIRST LETTER OF EACH WORD).

59) WRITE A PROGRAM TO EXTRACT A SUBSTRING BETWEEN TWO SPECIFIED CHARACTERS.

60) WRITE A PROGRAM TO FIND THE LONGEST PALINDROMIC SUBSTRING IN A GIVEN STRING.

61) WRITE A PROGRAM TO FIND ALL SUBSTRINGS OF A STRING.

62) WRITE A PROGRAM TO ROTATE A STRING BY A GIVEN NUMBER OF POSITIONS.

63) WRITE A PROGRAM TO INSERT A CHARACTER AFTER EVERY N CHARACTERS IN A STRING.

64) WRITE A PROGRAM TO CHECK IF A STRING CAN BE REARRANGED TO FORM A PALINDROME.

65) WRITE A PROGRAM TO CALCULATE THE ASCII DIFFERENCE BETWEEN CONSECUTIVE CHARACTERS IN A STRING.

66) WRITE A PROGRAM TO FIND THE POSITION OF A SUBSTRING IN A STRING.

67) WRITE A PROGRAM TO CONVERT A STRING INTO A LIST OF WORDS.

68) WRITE A PROGRAM TO REMOVE MULTIPLE SPACES BETWEEN WORDS IN A STRING.

69) WRITE A PROGRAM TO FIND THE SMALLEST AND LARGEST WORD IN A STRING.

70) WRITE A PROGRAM TO CONVERT A STRING TO CAMEL CASE.

71) WRITE A PROGRAM TO CONVERT A STRING TO SNAKE CASE.

72) WRITE A PROGRAM TO CONVERT A STRING TO KEBAB CASE.

73) WRITE A PROGRAM TO COUNT UPPERCASE AND LOWERCASE LETTERS IN A STRING.

74) WRITE A PROGRAM TO FIND THE LONGEST REPEATING CHARACTER SEQUENCE IN A STRING.

75) WRITE A PROGRAM TO CHECK IF A STRING CONSISTS OF ONLY UPPERCASE OR ONLY LOWERCASE LETTERS.

76) WRITE A PROGRAM TO COUNT THE NUMBER OF UPPERCASE, LOWERCASE, DIGIT, AND SPECIAL CHARACTERS IN A STRING.

77) WRITE A PROGRAM TO FIND THE SHORTEST PALINDROMIC SUBSTRING IN A STRING.

78) WRITE A PROGRAM TO REPLACE ALL SPACES IN A STRING WITH HYPHENS.

79) WRITE A PROGRAM TO CONCATENATE TWO STRINGS ALTERNATELY CHARACTER BY CHARACTER.

80) WRITE A PROGRAM TO REMOVE ALL CHARACTERS THAT APPEAR MORE THAN ONCE IN A STRING.

81) WRITE A PROGRAM TO MASK ALL BUT THE LAST FOUR CHARACTERS OF A STRING.

82) WRITE A PROGRAM TO CONVERT A STRING WITH UNDERSCORES TO "TITLE CASE."

83) WRITE A PROGRAM TO COUNT HOW MANY TIMES EACH WORD OCCURS IN A STRING.

84) WRITE A PROGRAM TO CHECK IF ALL WORDS IN A STRING ARE OF THE SAME LENGTH.

85) WRITE A PROGRAM TO FIND ALL WORDS IN A STRING THAT START WITH A SPECIFIED LETTER.

86) WRITE A PROGRAM TO REVERSE THE ORDER OF WORDS IN A STRING.

87) WRITE A PROGRAM TO CAPITALIZE THE FIRST AND LAST LETTERS OF EACH WORD IN A STRING.

88) WRITE A PROGRAM TO REMOVE ALL DUPLICATE WORDS FROM A STRING.

89) WRITE A PROGRAM TO DISPLAY ALL POSSIBLE COMBINATIONS OF TWO WORDS FROM A STRING.

90) WRITE A PROGRAM TO CREATE A NEW STRING BY COMBINING CHARACTERS AT ODD INDICES OF TWO STRINGS.

91) WRITE A PROGRAM TO CHECK IF A STRING IS A PANAGRAM (CONTAINS ALL LETTERS OF THE ALPHABET).

92) WRITE A PROGRAM TO COUNT THE NUMBER OF TIMES A CHARACTER APPEARS IN A STRING.

93) WRITE A PROGRAM TO SPLIT A STRING BASED ON MULTIPLE DELIMITERS.

94) WRITE A PROGRAM TO REPLACE THE FIRST AND LAST OCCURRENCE OF A CHARACTER IN A STRING.

95) WRITE A PROGRAM TO ENCODE A STRING USING RUN-LENGTH ENCODING.

96) WRITE A PROGRAM TO DECODE A RUN-LENGTH ENCODED STRING.

97) WRITE A PROGRAM TO FIND ALL SUBSEQUENCES OF A STRING.

98) WRITE A PROGRAM TO FIND THE FIRST REPEATED WORD IN A STRING.

99) WRITE A PROGRAM TO REMOVE A GIVEN WORD FROM A STRING.

100) WRITE A PROGRAM TO REPLACE MULTIPLE OCCURRENCES OF A WORD WITH A SINGLE INSTANCE.

101) WRITE A PROGRAM TO REVERSE ONLY THE VOWELS IN A STRING.

102) WRITE A PROGRAM TO SORT WORDS IN A STRING BASED ON LENGTH.

103) WRITE A PROGRAM TO SPLIT A STRING INTO SUBSTRINGS OF EQUAL LENGTH.

104) WRITE A PROGRAM TO ADD A CHARACTER AT EVERY SPECIFIED POSITION IN A STRING.

105) WRITE A PROGRAM TO REVERSE WORDS AT EVEN INDEX POSITIONS IN A STRING.

106) WRITE A PROGRAM TO FIND ALL POSSIBLE PALINDROMIC SUBSTRINGS IN A STRING.

107) WRITE A PROGRAM TO REPLACE THE MIDDLE CHARACTER IN EACH WORD WITH AN ASTERISK.

108) WRITE A PROGRAM TO COUNT WORDS THAT START AND END WITH THE SAME CHARACTER.

109) WRITE A PROGRAM TO CHECK IF A STRING CAN BE SPLIT INTO TWO IDENTICAL HALVES.

110) WRITE A PROGRAM TO DISPLAY ALL STRINGS THAT CAN BE GENERATED BY INSERTING ONE CHARACTER.

111) WRITE A PROGRAM TO CHECK IF A STRING IS AN ABBREVIATION OF ANOTHER STRING.

112) WRITE A PROGRAM TO FIND THE LONGEST PREFIX THAT IS ALSO A SUFFIX.

113) WRITE A PROGRAM TO CHECK IF A STRING IS A PERFECT REPEATED SUBSTRING.

114) WRITE A PROGRAM TO FIND ALL UNIQUE WORDS THAT CAN BE FORMED BY REARRANGING CHARACTERS.

115) WRITE A PROGRAM TO CHECK IF ALL CHARACTERS IN A STRING HAVE EVEN FREQUENCY.

116) WRITE A PROGRAM TO ROTATE EACH WORD IN A STRING BY N POSITIONS.

117) WRITE A PROGRAM TO CREATE A STRING FROM THE ASCII VALUES OF A NUMBER STRING.

118) WRITE A PROGRAM TO FIND THE KTH LARGEST CHARACTER IN A STRING.

119) WRITE A PROGRAM TO CONVERT A STRING WITH LETTERS TO CORRESPONDING PHONE KEYS.

120) WRITE A PROGRAM TO GENERATE A STRING BY REMOVING SPECIFIED CHARACTERS.

121) WRITE A PROGRAM TO FORM ALL PALINDROMIC WORDS IN A STRING.

122) WRITE A PROGRAM TO CHECK IF A STRING CAN BE SPLIT INTO VALID WORDS (WORD DICTIONARY CHECK).

123) WRITE A PROGRAM TO FORM A NEW STRING BY DOUBLING EACH VOWEL.

124) WRITE A PROGRAM TO CHECK IF ONE STRING CAN BE TRANSFORMED INTO ANOTHER BY SINGLE CHARACTER SWAPS.

125) WRITE A PROGRAM TO FIND WORDS THAT ARE SUBSTRINGS OF OTHER WORDS IN A STRING.

126) WRITE A PROGRAM TO COUNT PALINDROMIC SUBSEQUENCES IN A STRING.

127) WRITE A PROGRAM TO EXTRACT ALTERNATE WORDS FROM A STRING.

128) WRITE A PROGRAM TO REMOVE ALTERNATE CHARACTERS FROM A STRING.

129) WRITE A PROGRAM TO COUNT VOWELS IN EACH WORD OF A STRING.

130) WRITE A PROGRAM TO INTERCHANGE THE FIRST AND LAST CHARACTER OF EACH WORD IN A STRING.

131) WRITE A PROGRAM TO CHECK IF A STRING IS A VALID ANAGRAM OF A PALINDROME.

132) WRITE A PROGRAM TO FIND ALL CHARACTERS THAT APPEAR MORE THAN ONCE IN A STRING.

133) WRITE A PROGRAM TO COUNT THE NUMBER OF UNIQUE CHARACTERS IN A STRING.

134) WRITE A PROGRAM TO REPLACE ALL DIGITS IN A STRING WITH A SPECIFIED CHARACTER.

135) WRITE A PROGRAM TO FIND THE FIRST AND LAST VOWEL IN A STRING.

136) WRITE A PROGRAM TO REMOVE LEADING AND TRAILING WHITESPACE FROM A STRING.

137) WRITE A PROGRAM TO COUNT HOW MANY TIMES EACH VOWEL OCCURS IN A STRING.

138) WRITE A PROGRAM TO CHECK IF A STRING CAN BE REARRANGED TO FORM A SPECIFIED STRING.

139) WRITE A PROGRAM TO REMOVE ALL CHARACTERS IN A STRING THAT ARE NOT ALPHANUMERIC.

140) WRITE A PROGRAM TO DISPLAY THE SECOND LARGEST CHARACTER IN A STRING.

141) WRITE A PROGRAM TO REPLACE EACH SPACE IN A STRING WITH A SPECIFIED CHARACTER.

142) WRITE A PROGRAM TO FIND THE LENGTH OF THE LONGEST WORD IN A STRING.

143) WRITE A PROGRAM TO COUNT THE NUMBER OF SENTENCES IN A STRING.

144) WRITE A PROGRAM TO CREATE A STRING FROM THE FIRST LETTER OF EACH WORD IN A STRING.

145) WRITE A PROGRAM TO CHECK IF A STRING IS A VALID IP ADDRESS.

146) WRITE A PROGRAM TO GENERATE ALL SUBSTRING COMBINATIONS OF A STRING.

147) WRITE A PROGRAM TO CHECK IF A STRING IS A VALID HEXADECIMAL NUMBER.

148) WRITE A PROGRAM TO FIND ALL PAIRED CHARACTERS IN A STRING.

149) WRITE A PROGRAM TO CONVERT A STRING TO A LIST OF CHARACTERS.

150) WRITE A PROGRAM TO REPLACE ALL OCCURRENCES OF A WORD IN A STRING WITH ANOTHER WORD.

=> 8. NUMBER PROGRAMS

1) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS PRIME NUMBER OR NOT.

2) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS PALINDROME NUMBER OR NOT.

3) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER PERFECT OR ABUNDANT OR DEFICIENT NUMBER OR NOT.

4) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS PRONIC NUMBER OR NOT.

5) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS NEON NUMBER OR NOT.

6) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS NIVEN NUMBER (HARSHAD NUMBER) OR NOT.

7) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS ARMSTRONG NUMBER OR NOT.

8) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS ISBN NUMBER OR NOT.

9) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS EMIRP NUMBER OR NOT.

10) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS PRIME PALINDROME NUMBER OR NOT.

11) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS DISARIUM NUMBER OR NOT.

12) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS NARCISSISTIC NUMBER OR NOT.

13) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS AUTOMORPHIC NUMBER OR NOT.

14) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS FASCINATING NUMBER OR NOT.

15) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS REVERSE PRIME NUMBER OR NOT.

16) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS TWIN PRIME NUMBER OR NOT.

17) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS BUZZ NUMBER OR NOT.

18) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS EUCLID NUMBER OR NOT.

19) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS UNUSUAL NUMBER OR NOT.

20) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS ANTI PRIME NUMBER (HIGHLY COMPOSITE NUMBER) OR NOT.

21) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS MAGIC NUMBER OR NOT.

22) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS WEIRD NUMBER OR NOT.

23) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS MERSENNE PRIME NUMBER OR NOT.

24) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS STRANGE NUMBER OR NOT.

25) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS MYSTERY NUMBER OR NOT.

26) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS PERFECT SQUARE.

27) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS HAPPY NUMBER OR NOT.

28) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS SEMI-PRIME NUMBER OR NOT.

29) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS KAPREKAR NUMBER OR NOT.

30) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS PSEUDOPERFECT NUMBER OR NOT.

31) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS TRIANGULAR NUMBER OR NOT.

32) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS FIBONACCI NUMBER OR NOT.

33) WRITE A PROGRAM TO CHECK WHETHER A GIVEN NUMBER BY USER IS UNIQUE NUMBER OR NOT.

34) WRITE A PROGRAM TO CHECK IF A NUMBER IS AN EIGHTH POWER.

35) WRITE A PROGRAM TO CHECK IF A NUMBER IS A FACTORIAL NUMBER.

36. WRITE A PROGRAM TO CHECK IF A NUMBER IS A HAMMING NUMBER.

37. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A LYCHREL NUMBER.

38. WRITE A PROGRAM TO CHECK IF A NUMBER IS A SQUARE-FREE NUMBER.

39. WRITE A PROGRAM TO CHECK IF A NUMBER IS A FERMAT PRIME.

40. WRITE A PROGRAM TO CHECK IF A NUMBER IS A HEXAGONAL NUMBER.

41. WRITE A PROGRAM TO CHECK IF A NUMBER IS A CENTERED TRIANGULAR NUMBER.

42. WRITE A PROGRAM TO FIND THE NEXT HIGHER PRIME NUMBER AFTER A GIVEN NUMBER.

43. WRITE A PROGRAM TO GENERATE A LIST OF ALL MERSENNE PRIMES UP TO A GIVEN NUMBER.

44. WRITE A PROGRAM TO CHECK IF A NUMBER IS A REPDIGIT (ALL DIGITS ARE THE SAME).

45. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A STROBOGRAMMATIC NUMBER.

46. WRITE A PROGRAM TO CHECK IF A NUMBER IS A TRIANGULAR PENTAGONAL HEXAGONAL NUMBER.

47. WRITE A PROGRAM TO CHECK IF A NUMBER IS A SMITH NUMBER.

48. WRITE A PROGRAM TO CALCULATE THE DIGITAL ROOT OF A NUMBER.

49. WRITE A PROGRAM TO FIND THE SECOND LARGEST DIGIT IN A NUMBER.

50. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A KEITH NUMBER.

51. WRITE A PROGRAM TO CHECK IF A NUMBER IS A CARMICHAEL NUMBER.

52. WRITE A PROGRAM TO CHECK IF A NUMBER IS A BELL NUMBER.

53. WRITE A PROGRAM TO CHECK IF A NUMBER IS A THABIT NUMBER.

54. WRITE A PROGRAM TO FIND THE FIRST N FERMAT NUMBERS.

55. WRITE A PROGRAM TO CHECK IF A NUMBER IS A SPHENIC NUMBER (PRODUCT OF THREE DISTINCT PRIMES).

56. WRITE A PROGRAM TO FIND THE NTH MOTZKIN NUMBER.

57. WRITE A PROGRAM TO CHECK IF A NUMBER IS A "HAPPY PRIME" (PRIME AND HAPPY).

58. WRITE A PROGRAM TO CHECK IF A NUMBER IS A SOCIABLE NUMBER.

59. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS AN ACHILLES NUMBER.

60. WRITE A PROGRAM TO CALCULATE THE NTH HEXAGONAL NUMBER.

61. WRITE A PROGRAM TO CHECK IF A NUMBER IS A LATTICE NUMBER.

62. WRITE A PROGRAM TO CHECK IF A NUMBER IS A PALINDROMIC SQUARE.

63. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A DE POLIGNAC NUMBER.

64. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A SUPERPRIME.

65. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A SOPHIE GERMAIN PRIME.

66. WRITE A PROGRAM TO FIND THE NTH HARMONIC NUMBER.

67. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A STRONG NUMBER.

68. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS AN AMICABLE NUMBER.

69. WRITE A PROGRAM TO CALCULATE THE NTH LUCAS NUMBER.

70. WRITE A PROGRAM TO CHECK IF A NUMBER IS A PARABOLIC NUMBER.

71. WRITE A PROGRAM TO CHECK IF A NUMBER IS A PSEUDOPRIME.

72. WRITE A PROGRAM TO FIND THE NEAREST PERFECT SQUARE GREATER THAN A GIVEN NUMBER.

73. WRITE A PROGRAM TO CHECK IF A NUMBER IS A RAMANUJAN NUMBER.

74. WRITE A PROGRAM TO FIND THE NTH SQUARE PYRAMIDAL NUMBER.

75. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A SEMIPERFECT NUMBER.

76. WRITE A PROGRAM TO CHECK IF A NUMBER IS A POLITE NUMBER.

77. WRITE A PROGRAM TO CALCULATE THE NTH CULLEN NUMBER.

78. WRITE A PROGRAM TO CHECK IF A NUMBER IS AN AMICABLE PAIR.

79. WRITE A PROGRAM TO FIND ALL NUMBERS THAT ARE NOT THE SUM OF TWO ABUNDANT NUMBERS.

80. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS AN EXTRAVERTED NUMBER.

81. WRITE A PROGRAM TO CHECK IF A NUMBER IS A TETRANACCI NUMBER.

82. WRITE A PROGRAM TO FIND THE NEAREST DEFICIENT NUMBER LESS THAN A GIVEN NUMBER.

83. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A DIGITAL INVARIANT.

84. WRITE A PROGRAM TO CHECK IF A NUMBER IS A UNIQUE PRIME FACTOR OF ANOTHER.

85. WRITE A PROGRAM TO FIND THE SUM OF ALL PERFECT SQUARES UP TO A GIVEN NUMBER.

86. WRITE A PROGRAM TO FIND ALL PRIME NUMBERS UP TO A GIVEN NUMBER USING THE SIEVE OF ERATOSTHENES.

87. WRITE A PROGRAM TO CALCULATE THE NUMBER OF DIVISORS OF A NUMBER.

88. WRITE A PROGRAM TO GENERATE THE FIBONACCI SEQUENCE UP TO A GIVEN NUMBER.

89. WRITE A PROGRAM TO DETERMINE IF A NUMBER IS A FACTORIAL PRIME.

90. WRITE A PROGRAM TO FIND ALL THE NUMBERS UP TO N THAT ARE DIVISIBLE BY THE SUM OF THEIR DIGITS.

91. WRITE A PROGRAM TO FIND THE NEXT HAPPY NUMBER AFTER A GIVEN NUMBER.

92. WRITE A PROGRAM TO GENERATE THE CATALAN NUMBERS UP TO A GIVEN TERM.

93. WRITE A PROGRAM TO FIND ALL CIRCULAR PRIMES UP TO A GIVEN NUMBER.

94. WRITE A PROGRAM TO FIND THE NEAREST PALINDROME GREATER THAN A GIVEN NUMBER.

95. WRITE A PROGRAM TO CHECK IF A NUMBER IS A REPUNIT NUMBER (ALL ONES IN BASE 10).

96. WRITE A PROGRAM TO FIND THE NUMBER OF ORDERED PAIRS OF NUMBERS (A, B) SUCH THAT $A * B$ EQUALS A GIVEN NUMBER.

97. WRITE A PROGRAM TO GENERATE ALL UNIQUE PRIME FACTORS OF A NUMBER.

98. WRITE A PROGRAM TO FIND THE LONGEST SEQUENCE OF CONSECUTIVE PRIME NUMBERS THAT SUM UP TO A GIVEN NUMBER.

99. WRITE A PROGRAM TO FIND ALL PRIME NUMBERS IN A SPECIFIC RANGE THAT ARE ALSO EMIRP NUMBERS.

100. WRITE A PROGRAM TO FIND THE LARGEST PRIME FACTOR OF A GIVEN NUMBER.

101. WRITE A PROGRAM TO GENERATE ALL PYTHAGOREAN TRIPLETS UP TO A GIVEN LIMIT.

102. WRITE A PROGRAM TO DETERMINE IF A NUMBER CAN BE EXPRESSED AS THE SUM OF TWO OR MORE CONSECUTIVE POSITIVE INTEGERS.

103. WRITE A PROGRAM TO FIND THE MULTIPLICATIVE DIGITAL ROOT OF A NUMBER.

104. WRITE A PROGRAM TO FIND THE SMALLEST NUMBER WHOSE SUM OF DIGITS IS EQUAL TO A GIVEN NUMBER.

105. WRITE A PROGRAM TO FIND ALL NUMBERS LESS THAN A GIVEN NUMBER THAT ARE PALINDROMIC PRIMES.

106. WRITE A PROGRAM TO FIND THE MAXIMUM PRODUCT OF TWO ADJACENT DIGITS IN A NUMBER.

107. WRITE A PROGRAM TO CHECK IF A NUMBER CAN BE EXPRESSED AS A SUM OF TWO SQUARES.

108. WRITE A PROGRAM TO FIND THE SUM OF THE CUBES OF ALL DIGITS IN A NUMBER.

109. WRITE A PROGRAM TO FIND THE NUMBER OF TRAILING ZEROES IN THE FACTORIAL OF A GIVEN NUMBER.

110. WRITE A PROGRAM TO CHECK IF A NUMBER IS A MULTIPLE OF ANOTHER NUMBER.

111. WRITE A PROGRAM TO FIND THE SUM OF ALL DIGITS IN A NUMBER.

112. WRITE A PROGRAM TO COUNT THE NUMBER OF DIGITS IN A NUMBER.

113. WRITE A PROGRAM TO CALCULATE THE PRODUCT OF DIGITS IN A NUMBER.

114. WRITE A PROGRAM TO REVERSE THE DIGITS OF A NUMBER.

115. WRITE A PROGRAM TO CHECK IF A NUMBER IS A SINGLE-DIGIT OR MULTI-DIGIT NUMBER.

116. WRITE A PROGRAM TO CALCULATE THE FACTORIAL OF A NUMBER USING ITERATION.

117. WRITE A PROGRAM TO CHECK IF A NUMBER IS WITHIN A SPECIFIC RANGE.

118. WRITE A PROGRAM TO CALCULATE THE SUM OF THE SQUARES OF THE FIRST N NATURAL NUMBERS.

119. WRITE A PROGRAM TO FIND THE SUM OF THE FIRST N PRIME NUMBERS.

120. WRITE A PROGRAM TO FIND THE SUM OF THE FIRST N FIBONACCI NUMBERS.

121. WRITE A PROGRAM TO COUNT THE OCCURRENCES OF A SPECIFIC DIGIT IN A NUMBER.

122. WRITE A PROGRAM TO FIND THE DIFFERENCE BETWEEN THE SUM OF SQUARES AND THE SQUARE OF SUMS OF THE FIRST N NATURAL NUMBERS.

=> 9. NUMBER SYSTEMS

1) WRITE A PROGRAM TO CONVERT USER-INPUT BINARY NUMBER TO OCTAL NUMBER AND DISPLAY IT.

2) WRITE A PROGRAM TO CONVERT USER-INPUT BINARY NUMBER TO DECIMAL NUMBER AND DISPLAY IT.

3) WRITE A PROGRAM TO CONVERT USER-INPUT BINARY NUMBER TO HEXADECIMAL NUMBER AND DISPLAY IT.

4) WRITE A PROGRAM TO CONVERT USER-INPUT OCTAL NUMBER TO BINARY NUMBER AND DISPLAY IT.

5) WRITE A PROGRAM TO CONVERT USER-INPUT OCTAL NUMBER TO DECIMAL NUMBER AND DISPLAY IT.

6) WRITE A PROGRAM TO CONVERT USER-INPUT OCTAL NUMBER TO HEXADECIMAL NUMBER AND DISPLAY IT.

7) WRITE A PROGRAM TO CONVERT USER-INPUT DECIMAL NUMBER TO BINARY NUMBER AND DISPLAY IT.

8) WRITE A PROGRAM TO CONVERT USER-INPUT DECIMAL NUMBER TO OCTAL NUMBER AND DISPLAY IT.

9) WRITE A PROGRAM TO CONVERT USER-INPUT DECIMAL NUMBER TO HEXADECIMAL NUMBER AND DISPLAY IT.

10) WRITE A PROGRAM TO CONVERT USER-INPUT HEXADECIMAL NUMBER TO BINARY NUMBER AND DISPLAY IT.

11) WRITE A PROGRAM TO CONVERT USER-INPUT HEXADECIMAL NUMBER TO OCTAL NUMBER AND DISPLAY IT.

12) WRITE A PROGRAM TO CONVERT USER-INPUT HEXADECIMAL NUMBER TO DECIMAL NUMBER AND DISPLAY IT.

=> 10. FUNCTIONS/METHODS

1) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A GREETING MESSAGE.

2) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE CURRENT DATE AND TIME.

3) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A RANDOM NUMBER BETWEEN 1 AND 100.

4) WRITE A FUNCTION THAT ACCEPTS TWO NUMBERS AS ARGUMENTS AND DISPLAYS THEIR SUM.

5) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE CURRENT DAY OF THE WEEK.

6) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS ALL EVEN NUMBERS FROM 1 TO 50.

7) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE FACTORIAL OF A GIVEN NUMBER.

8) WRITE A FUNCTION THAT ACCEPTS A NUMBER AS AN ARGUMENT AND PRINTS ALL ITS DIVISORS.

9) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A LIST OF 10 RANDOM INTEGERS.

10) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE SMALLEST ELEMENT IN A PREDEFINED LIST.

11) WRITE A FUNCTION THAT ACCEPTS A STRING AS AN ARGUMENT AND RETURNS THE NUMBER OF VOWELS IN THE STRING.

12) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE MULTIPLICATION TABLE OF 5.

13) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A COUNTDOWN FROM 10 TO 1.

14) WRITE A FUNCTION THAT TAKES A LIST AS AN ARGUMENT AND PRINTS EACH ELEMENT ON A NEW LINE.

15) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A RANDOM ALPHANUMERIC CHARACTER.

16) WRITE A FUNCTION THAT TAKES A CHARACTER AS AN ARGUMENT AND RETURNS TRUE IF IT IS UPPERCASE.

17) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE FIRST 20 FIBONACCI NUMBERS.

18) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE SQUARE ROOT OF 25.

19) WRITE A FUNCTION THAT TAKES A STRING AS AN ARGUMENT AND RETURNS THE SAME STRING IN REVERSE ORDER.

20) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A 3X3 MATRIX WITH RANDOM VALUES.

21) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS NUMBERS FROM 1 TO 100, SKIPPING MULTIPLES OF 5.

22) WRITE A FUNCTION THAT TAKES A LIST OF NUMBERS AS AN ARGUMENT AND RETURNS THE LIST SORTED IN ASCENDING ORDER.

23) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A TRIANGLE OF STARS.

24) WRITE A FUNCTION THAT ACCEPTS A NUMBER AS AN ARGUMENT AND RETURNS WHETHER IT IS A PRIME NUMBER.

25) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE ASCII VALUES OF CHARACTERS FROM 'A' TO 'Z'.

26) WRITE A FUNCTION THAT TAKES TWO STRINGS AS ARGUMENTS AND RETURNS THE LONGEST OF THE TWO.

27) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE CUBE OF EACH NUMBER FROM 1 TO 10.

28) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A LIST OF THE FIRST 10 PRIME NUMBERS.

29) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE NUMBER OF DAYS IN THE CURRENT MONTH.

30) WRITE A FUNCTION THAT TAKES A NUMBER AS AN ARGUMENT AND RETURNS THE SQUARE OF THE NUMBER.

31) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS ALL EVEN NUMBERS BETWEEN 50 AND 100.

32) WRITE A FUNCTION THAT TAKES A STRING AS AN ARGUMENT AND RETURNS THE COUNT OF UPPERCASE LETTERS.

33) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE SUM OF THE FIRST 50 NATURAL NUMBERS.

34) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A RECTANGLE OF STARS.

35) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE GREATEST COMMON DIVISOR (GCD) OF TWO NUMBERS.

36) WRITE A FUNCTION THAT TAKES A LIST OF NUMBERS AS AN ARGUMENT AND RETURNS THE LARGEST NUMBER IN THE LIST.

37) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A 5X5 IDENTITY MATRIX.

38) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE REVERSE OF A GIVEN INTEGER.

39) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS AN ISOSCELES TRIANGLE OF STARS.

40) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A RANDOM 6-DIGIT PASSWORD.

41) WRITE A FUNCTION THAT TAKES A NUMBER AS AN ARGUMENT AND RETURNS TRUE IF IT IS A PALINDROME.

42) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS NUMBERS DIVISIBLE BY 3 BETWEEN 1 AND 100.

43) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE FIRST 10 TERMS OF AN ARITHMETIC SEQUENCE.

44) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE BINARY REPRESENTATION OF NUMBERS FROM 1 TO 10.

45) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE SUM OF ALL EVEN DIGITS IN A GIVEN INTEGER.

46) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE LENGTH OF THE LONGEST WORD IN A SENTENCE.

47) WRITE A FUNCTION THAT ACCEPTS A TEMPERATURE IN CELSIUS AS AN ARGUMENT AND CONVERTS IT TO FAHRENHEIT.

48) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE FIRST 5 TRIANGULAR NUMBERS.

49) WRITE A FUNCTION THAT ACCEPTS A POSITIVE INTEGER AND RETURNS THE SUM OF ITS DIGITS.

50) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS AN HOURGLASS PATTERN OF STARS.

51) WRITE A FUNCTION THAT ACCEPTS TWO DATES AS ARGUMENTS AND RETURNS THE NUMBER OF DAYS BETWEEN THEM.

52) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT FINDS AND RETURNS THE LARGEST PRIME FACTOR OF A NUMBER.

53) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A 4X4 SPIRAL MATRIX.

54) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS NUMBERS IN A DIAMOND PATTERN.

55) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A RANDOM ALPHANUMERIC PASSWORD OF LENGTH 12.

56) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A SORTED LIST OF UNIQUE WORDS IN A SENTENCE.

57) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE LUCAS SEQUENCE UP TO 10 TERMS.

58) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS A NEW LIST WITH DUPLICATES REMOVED.

59) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE MOST FREQUENT CHARACTER IN A STRING.

60) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A HALF-PYRAMID OF ALTERNATING 0S AND 1S.

61) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE FACTORIAL OF A NUMBER USING RECURSION.

62) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A RANDOM 4X4 MATRIX AND DISPLAYS ITS TRANSPOSE.

63) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE FIRST 10 TERMS OF THE CATALAN SEQUENCE.

64) WRITE A FUNCTION THAT ACCEPTS TWO LISTS AND RETURNS A LIST CONTAINING THE INTERSECTION OF BOTH.

65) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE CALENDAR FOR THE CURRENT MONTH.

66) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS ITS BINARY EQUIVALENT.

67) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A RANDOM 3X3 MATRIX OF 0S AND 1S.

68) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS THE MODE (MOST COMMON VALUE).

69) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A 3X3 IDENTITY MATRIX USING RECURSION.

70) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A SHUFFLED VERSION OF A PREDEFINED LIST.

71) WRITE A FUNCTION THAT TAKES A STRING AS INPUT AND RETURNS IT WITH DUPLICATE CHARACTERS REMOVED.

72) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A STAIRCASE PATTERN WITH 5 STEPS.

73) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT CALCULATES THE PERIMETER OF A CIRCLE WITH A FIXED RADIUS.

74) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES AND DISPLAYS PASCAL'S TRIANGLE.

75) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS THE FREQUENCY OF EACH CHARACTER.

76) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE FIRST 20 TERMS OF THE PELL SEQUENCE.

77) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A MATRIX OF RANDOM FLOATING-POINT NUMBERS.

78) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND CHECKS IF IT IS A POWER OF 2.

79) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A PYRAMID OF NUMBERS.

80) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT FINDS AND DISPLAYS ALL ARMSTRONG NUMBERS UP TO 500.

81) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS ITS PRIME FACTORS AS A LIST.

82) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE LIST OF PERFECT SQUARES UP TO 1,000.

83) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE FIRST 100 PRIME NUMBERS IN A SPIRAL MATRIX.

84) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES AND DISPLAYS THE SIERPINSKI TRIANGLE.

85) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT FINDS AND PRINTS ALL TWIN PRIMES BELOW 1,000.

86) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND FINDS ITS DIGITAL ROOT RECURSIVELY.

87) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE COLLATZ SEQUENCE FOR A GIVEN NUMBER.

88) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT FINDS AND DISPLAYS ALL PYTHAGOREAN TRIPLES BELOW 100.

89) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES AND DISPLAYS THE HILBERT CURVE PATTERN.

90) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT CALCULATES AND DISPLAYS THE FIBONACCI SEQUENCE UP TO 1,000.

91) WRITE A FUNCTION THAT ACCEPTS A LIST OF POINTS AND RETURNS THE CONVEX HULL OF THOSE POINTS.

92) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT FINDS ALL NARCISSISTIC NUMBERS FROM 1 TO 10,000.

93) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS A LIST OF ITS DIVISORS.

94) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT CALCULATES AND PRINTS THE MANDELBROT SET.

95) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A KOCH SNOWFLAKE PATTERN.

96) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT CALCULATES AND PRINTS ALL MERSENNE PRIMES UP TO 100.

97) WRITE A FUNCTION THAT ACCEPTS AN INTEGER AND RETURNS TRUE IF IT IS A PERFECT NUMBER.

98) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS ALL ABUNDANT NUMBERS BELOW 1,000.

99) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE LIST OF SELF-DESCRIPTIVE NUMBERS.

100) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS THE NTH LARGEST NUMBER.

101) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE SQUARE OF ALL NUMBERS FROM 1 TO 20.

102) WRITE A FUNCTION THAT ACCEPTS TWO NUMBERS AND PRINTS WHETHER THE FIRST IS A MULTIPLE OF THE SECOND.

103) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE ASCII CODE OF A GIVEN CHARACTER.

104) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE SUM OF ALL ODD NUMBERS FROM 1 TO 50.

105) WRITE A FUNCTION THAT ACCEPTS A STRING AS AN ARGUMENT AND RETURNS TRUE IF IT IS A PALINDROME.

106) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS EACH LETTER OF THE ALPHABET IN UPPERCASE.

107) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A RANDOM FLOAT BETWEEN 0 AND 1.

108) WRITE A FUNCTION THAT ACCEPTS A TEMPERATURE IN FAHRENHEIT AND CONVERTS IT TO CELSIUS.

109) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A LIST OF 5 RANDOM WORDS.

110) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A SEQUENCE OF NUMBERS IN A CHECKERBOARD PATTERN.

111) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE REVERSE OF EACH NUMBER FROM 1 TO 10.

112) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE PRODUCT OF ALL INTEGERS IN A PREDEFINED LIST.

113) WRITE A FUNCTION THAT ACCEPTS A CHARACTER AS AN ARGUMENT AND RETURNS TRUE IF IT IS A DIGIT.

114) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE SUM OF SQUARES FROM 1 TO 20.

115) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS ALL THE EVEN-INDEXED CHARACTERS OF A STRING.

116) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE PRODUCT OF THE FIRST 10 EVEN NUMBERS.

117) WRITE A FUNCTION THAT ACCEPTS A STRING AS AN ARGUMENT AND RETURNS THE NUMBER OF SPACES IN IT.

118) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE REVERSE OF A PREDEFINED STRING.

119) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A LIST OF 10 RANDOM LOWERCASE LETTERS.

120) WRITE A FUNCTION THAT TAKES A NAME AS AN ARGUMENT AND RETURNS IT WITH THE FIRST LETTER CAPITALIZED.

121) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE CUMULATIVE SUM OF NUMBERS FROM 1 TO 50.

122) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A RANDOM HEXADECIMAL COLOR CODE.

123) WRITE A FUNCTION THAT TAKES A RADIUS AS AN ARGUMENT AND RETURNS THE CIRCUMFERENCE OF A CIRCLE.

124) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A SQUARE OF NUMBERS FROM 1 TO 9.

125) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A RANDOM CHOICE FROM A PREDEFINED LIST OF FRUITS.

126) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS TRUE IF IT STARTS WITH A VOWEL.

127) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS ALL MULTIPLES OF 7 BETWEEN 1 AND 100.

128) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A RANDOM 3-DIGIT NUMBER.

129) WRITE A FUNCTION THAT TAKES A NUMBER AS AN ARGUMENT AND RETURNS THE NUMBER OF DIGITS IT CONTAINS.

130) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE ALPHABET IN REVERSE ORDER.

131) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE CURRENT TIME IN “HH:MM” FORMAT.

132) WRITE A FUNCTION THAT ACCEPTS TWO NUMBERS AND RETURNS THEIR LEAST COMMON MULTIPLE (LCM).

133) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS NUMBERS IN PAIRS (1,2), (3,4), ETC., UP TO 20.

134) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A RANDOM USERNAME.

135) WRITE A FUNCTION THAT ACCEPTS A WORD AND RETURNS TRUE IF IT IS AN ANAGRAM OF "LISTEN".

136) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A LIST OF COUNTRIES STARTING WITH THE LETTER "B".

137) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS ALL ODD NUMBERS BETWEEN 100 AND 200.

138) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE NUMBER OF DAYS LEFT IN THE CURRENT YEAR.

139) WRITE A FUNCTION THAT ACCEPTS A YEAR AND RETURNS TRUE IF IT IS A LEAP YEAR.

140) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE FIRST 10 EVEN NUMBERS.

141) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE UPPERCASE VERSION OF A GIVEN SENTENCE.

142) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS THE SECOND LARGEST NUMBER.

143) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A 3X3 GRID OF INCREMENTING NUMBERS.

144) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A SQUARE PATTERN OF RANDOM SYMBOLS.

145) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS NUMBERS IN A DIAMOND PATTERN.

146) WRITE A FUNCTION THAT ACCEPTS A LIST OF WORDS AND RETURNS A NEW LIST WITH THE LENGTHS OF EACH WORD.

147) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS ALL NUMBERS DIVISIBLE BY 3 FROM 1 TO 30.

148) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE LAST DIGIT OF A GIVEN INTEGER.

149) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A COUNTDOWN FROM 100 TO 0 BY TENS.

150) WRITE A FUNCTION THAT TAKES A LIST OF NUMBERS AND RETURNS TRUE IF THEY ARE IN ASCENDING ORDER.

151) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A SEQUENCE OF PERFECT SQUARES UP TO 200.

152) WRITE A FUNCTION THAT ACCEPTS A LIST OF STRINGS AND RETURNS THE LONGEST STRING.

153) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A LIST OF THE FIRST 50 FIBONACCI NUMBERS.

154) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE DIFFERENCE BETWEEN TWO RANDOM INTEGERS.

155) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE MULTIPLICATION TABLE FOR EACH NUMBER 1 TO 10.

156) WRITE A FUNCTION THAT ACCEPTS A BASE AND EXPONENT AND RETURNS THE RESULT.

157) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS ALL MULTIPLES OF 9 FROM 1 TO 100.

158) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS THE SAME STRING WITH EACH LETTER DUPLICATED.

159) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS ALL PRIME NUMBERS BELOW 100.

160) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE SUM OF ALL PRIME NUMBERS UP TO 500.

161) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A SEQUENCE OF TRIANGULAR NUMBERS.

162) WRITE A FUNCTION THAT TAKES A LIST OF NUMBERS AND RETURNS A LIST OF THOSE THAT ARE PRIME.

163) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A SEQUENCE OF PALINDROMIC NUMBERS.

164) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS NUMBERS IN A BINARY SEQUENCE.

165) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A HEXAGONAL PATTERN WITH RANDOM NUMBERS.

166) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS THE NUMBER OF CONSONANTS IN THE STRING.

167) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE FIRST 100 TERMS OF THE LUCAS SEQUENCE.

168) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS A LIST OF THEIR SQUARES.

169) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A LIST OF COMMON FACTORS OF TWO NUMBERS.

170) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS ALL TWO-DIGIT NUMBERS IN DESCENDING ORDER.

171) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A LIST OF 10 RANDOM FLOATING-POINT NUMBERS.

172) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS THE CLOSEST PRIME NUMBER.

173) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE SUM OF THE FIRST 50 ODD NUMBERS.

174) WRITE A FUNCTION THAT ACCEPTS A LIST OF WORDS AND RETURNS A LIST OF EACH WORD IN UPPERCASE.

175) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS ALL MULTIPLES OF 11 FROM 1 TO 100.

176) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE SUM OF ALL POSITIVE NUMBERS IN A LIST.

177) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A 5X5 MATRIX WITH A DIAGONAL LINE.

178) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS NUMBERS FROM 1 TO 50, ALTERNATING ODD AND EVEN.

179) WRITE A FUNCTION THAT ACCEPTS A LIST OF INTEGERS AND RETURNS A LIST WITH ONLY THE UNIQUE VALUES.

180) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT GENERATES A 5X5 IDENTITY MATRIX.

181) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE PRIME NUMBERS FROM 1 TO 1,000.

182) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS THE NUMBER OF TRAILING ZEROES IN ITS FACTORIAL.

183) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A LIST OF ALL MERSENNE PRIMES BELOW 10,000.

184) WRITE A FUNCTION THAT ACCEPTS A POSITIVE INTEGER AND RETURNS ITS DIGITAL ROOT USING RECURSION.

185) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A PATTERN OF CONCENTRIC CIRCLES OF NUMBERS.

186) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT GENERATES A LIST OF THE FIRST 100 TERMS OF THE FIBONACCI SEQUENCE.

187) WRITE A FUNCTION THAT ACCEPTS TWO NUMBERS AND RETURNS TRUE IF THEY ARE TWIN PRIMES.

188) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE FIRST 50 PERFECT NUMBERS.

189) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS A RANDOM MAZE PATTERN USING NUMBERS.

190) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS TRUE IF IT CONTAINS AT LEAST ONE DIGIT.

191) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A DIAGONAL PATTERN OF NUMBERS.

192) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE FACTORIAL OF NUMBERS FROM 1 TO 5.

193) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS TRUE IF IT IS A PRIME NUMBER.

194) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A CIRCLE PATTERN WITH ASTERISKS.

195) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A RANDOM HEXADECIMAL COLOR CODE.

196) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS A LIST OF SQUARES OF THOSE NUMBERS.

197) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS NUMBERS FROM 1 TO 100 THAT ARE DIVISIBLE BY 7 BUT NOT 3.

198) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS TRUE IF IT CONTAINS BOTH UPPERCASE AND LOWERCASE LETTERS.

199) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS THE CURRENT DATE IN “DD-MM-YYYY” FORMAT.

200) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A SIMPLE RECTANGLE PATTERN WITH STARS.

201) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS TRUE IF IT CONTAINS NO SPACES.

202) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A TRIANGLE OF STARS WITH HEIGHT 10.

203) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS TRUE IF THERE IS AT LEAST ONE EVEN NUMBER IN THE LIST.

204) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A NUMBER PYRAMID IN THE SHAPE OF A RHOMBUS.

205) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS THE GREATEST COMMON DIVISOR (GCD).

206) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE NUMBERS FROM 1 TO 100 IN A GRID OF 10x10.

207) WRITE A FUNCTION THAT ACCEPTS A LIST OF STRINGS AND RETURNS THE STRING WITH THE LEAST VOWELS.

208) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A STAR PATTERN IN THE SHAPE OF A HEART.

209) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE FIBONACCI SEQUENCE UP TO 100.

210) WRITE A FUNCTION THAT ACCEPTS A LIST OF STRINGS AND RETURNS A NEW LIST WITH THE FIRST AND LAST LETTERS OF EACH STRING.

211) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A RANDOM SENTENCE FROM A LIST OF PREDEFINED SENTENCES.

212) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS THE MULTIPLICATION TABLE OF A GIVEN NUMBER.

213) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS THE LARGEST NUMBER LESS THAN A GIVEN THRESHOLD.

214) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A RANDOM ELEMENT FROM A LIST OF COLORS.

215) WRITE A FUNCTION THAT ACCEPTS A STRING AND RETURNS TRUE IF IT IS A PALINDROME, IGNORING SPACES AND CASE.

216) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT DISPLAYS THE CURRENT TIME.

217) WRITE A FUNCTION THAT ACCEPTS A NUMBER AND RETURNS TRUE IF IT IS A PERFECT SQUARE.

218) WRITE A FUNCTION WITH NO ARGUMENTS AND NO RETURN TYPE THAT PRINTS A RIGHT-ANGLED TRIANGLE WITH STARS.

219) WRITE A FUNCTION THAT ACCEPTS A LIST OF NUMBERS AND RETURNS THE SUM OF THE SQUARES OF ALL EVEN NUMBERS.

220) WRITE A FUNCTION WITH NO ARGUMENTS AND A RETURN TYPE THAT RETURNS A RANDOM CITY NAME FROM A PREDEFINED LIST.

=> 11. DATE AND TIME

1. WRITE A PROGRAM TO DISPLAY THE CURRENT DATE AND TIME.

2. WRITE A PROGRAM TO PRINT THE DAY OF THE WEEK FOR A GIVEN DATE.

3. WRITE A PROGRAM TO SIMULATE A BASIC CLOCK DISPLAYING HOURS, MINUTES, AND SECONDS.

4. WRITE A PROGRAM TO IMPLEMENT A BASIC CALENDAR FUNCTION (DISPLAY MONTH AND YEAR).

5. WRITE A PROGRAM TO CONVERT 24-HOUR TIME TO 12-HOUR TIME FORMAT.

6. WRITE A PROGRAM TO DISPLAY THE CURRENT DATE IN DIFFERENT FORMATS (E.G., "YYYY-MM-DD", "DD-MM-YYYY").

7. WRITE A PROGRAM TO CALCULATE THE NUMBER OF DAYS BETWEEN TWO DATES.

8. WRITE A PROGRAM TO DISPLAY THE CURRENT TIME IN DIFFERENT TIME ZONES.

9. WRITE A PROGRAM TO FIND THE WEEK NUMBER FOR A GIVEN DATE.

10. WRITE A PROGRAM TO FIND OUT IF A GIVEN YEAR IS A LEAP YEAR OR NOT BASED ON DATE INPUT.

11. WRITE A PROGRAM TO DISPLAY ALL DATES OF A SPECIFIC DAY (E.G., SUNDAYS) IN A GIVEN MONTH AND YEAR.

12. WRITE A PROGRAM TO CALCULATE THE AGE IN YEARS, MONTHS, AND DAYS GIVEN A BIRTH DATE.

13. WRITE A PROGRAM TO DISPLAY A COUNTDOWN TIMER THAT UPDATES EVERY SECOND.

14. WRITE A PROGRAM TO ADD OR SUBTRACT A SPECIFIC NUMBER OF DAYS TO A GIVEN DATE AND DISPLAY THE RESULT.

15. WRITE A PROGRAM TO DISPLAY A FULL CALENDAR FOR A SPECIFIC YEAR, INCLUDING DAYS OF THE WEEK.

16. WRITE A PROGRAM TO CALCULATE THE DIFFERENCE IN HOURS, MINUTES, AND SECONDS BETWEEN TWO TIMES.

17. WRITE A PROGRAM TO CONVERT A GIVEN TIMESTAMP INTO A HUMAN-READABLE FORMAT (E.G., "3 HOURS AGO").

18. WRITE A PROGRAM TO CALCULATE THE NEXT OCCURRENCE OF A SPECIFIC DATE (E.G., NEXT FRIDAY) FROM TODAY.

19. WRITE A PROGRAM TO SIMULATE A STOPWATCH WITH START, STOP, AND RESET FUNCTIONALITY.

=> 12. OOP CONCEPTS

1) WRITE A PROGRAM TO CREATE A CLASS CALLED CAR WITH ATTRIBUTES LIKE MAKE, MODEL, AND YEAR, AND DISPLAY THEIR VALUES.

2) WRITE A PROGRAM TO CREATE A CLASS BOOK WITH ATTRIBUTES TITLE, AUTHOR, AND PRICE AND METHODS TO DISPLAY BOOK DETAILS.

3) WRITE A PROGRAM TO CREATE A CLASS CIRCLE WITH A METHOD TO CALCULATE AND RETURN ITS AREA BASED ON A GIVEN RADIUS.

4) WRITE A PROGRAM TO CREATE A CLASS EMPLOYEE WITH METHODS TO SET AND DISPLAY THE NAME AND SALARY OF AN EMPLOYEE.

5) WRITE A PROGRAM TO CREATE A CLASS RECTANGLE WITH METHODS TO CALCULATE AND DISPLAY THE PERIMETER OF A RECTANGLE BASED ON LENGTH AND WIDTH.

6) WRITE A PROGRAM TO CREATE A CLASS STUDENT WITH ATTRIBUTES NAME, ROLL_NUMBER, AND MARKS. IMPLEMENT A METHOD TO CHECK IF THE STUDENT HAS PASSED BASED ON MARKS.

7) WRITE A PROGRAM TO DEMONSTRATE INHERITANCE BY CREATING A BASE CLASS VEHICLE AND DERIVED CLASSES CAR AND BIKE WITH UNIQUE ATTRIBUTES.

8) WRITE A PROGRAM TO CREATE A CLASS ACCOUNT WITH METHODS TO DEPOSIT AND WITHDRAW MONEY, TRACKING BALANCE AND ALLOWING FOR OVERDRAFT CHECKS.

9) WRITE A PROGRAM TO DEMONSTRATE POLYMORPHISM BY CREATING A BASE CLASS SHAPE WITH A METHOD AREA() AND DERIVED CLASSES SQUARE AND TRIANGLE IMPLEMENTING AREA() ACCORDINGLY.

10) WRITE A PROGRAM TO CREATE A CLASS LIBRARY THAT MANAGES A LIST OF BOOKS, AND IMPLEMENT METHODS TO ADD, REMOVE, AND DISPLAY BOOKS IN THE LIBRARY.

11) WRITE A PROGRAM TO CREATE A CLASS HIERARCHY FOR AN EMPLOYEE MANAGEMENT SYSTEM WITH BASE CLASS EMPLOYEE AND DERIVED CLASSES MANAGER AND INTERN. INCLUDE A METHOD TO CALCULATE DIFFERENT SALARY STRUCTURES.

12) WRITE A PROGRAM TO CREATE AN ORDER MANAGEMENT SYSTEM WITH CLASSES ORDER, CUSTOMER, AND PRODUCT. INCLUDE METHODS TO PLACE ORDERS AND CALCULATE THE TOTAL COST.

13) WRITE A PROGRAM TO IMPLEMENT A BANKING SYSTEM WITH CLASSES ACCOUNT, SAVINGS_ACCOUNT, AND CURRENT_ACCOUNT. SHOW OVERLOADING OF METHODS FOR DIFFERENT ACCOUNT TYPES.

14) WRITE A PROGRAM TO SIMULATE A LIBRARY SYSTEM WITH CLASSES MEMBER, BOOK, AND LOAN. INCLUDE METHODS FOR BORROWING, RETURNING, AND CALCULATING DUE DATES.

15) WRITE A PROGRAM TO IMPLEMENT A MOVIE TICKET BOOKING SYSTEM WITH CLASSES MOVIE, SHOWTIME, AND BOOKING. INCLUDE METHODS FOR CHECKING AVAILABILITY, BOOKING SEATS, AND CANCELLING TICKETS.

16) WRITE A PROGRAM TO CREATE A CLASS POINT TO REPRESENT COORDINATES IN 2D SPACE AND INCLUDE A METHOD TO DISPLAY THE COORDINATES.

17) WRITE A PROGRAM TO CREATE A CLASS PERSON WITH A CONSTRUCTOR TO INITIALIZE NAME AND AGE, AND METHODS TO DISPLAY PERSON DETAILS.

18) WRITE A PROGRAM TO CREATE A CLASS TEMPERATURE_CONVERTER WITH METHODS TO CONVERT CELSIUS TO FAHRENHEIT AND VICE VERSA.

19) WRITE A PROGRAM TO CREATE A CLASS CALCULATOR WITH METHODS FOR BASIC OPERATIONS (ADD, SUBTRACT, MULTIPLY, DIVIDE) ON TWO NUMBERS.

20) WRITE A PROGRAM TO CREATE A CLASS COUNTER WITH AN INSTANCE VARIABLE THAT TRACKS THE NUMBER OF OBJECTS CREATED.

21) WRITE A PROGRAM TO DEMONSTRATE METHOD OVERRIDING BY CREATING A BASE CLASS ANIMAL WITH A METHOD SOUND(), AND DERIVED CLASSES DOG AND CAT THAT IMPLEMENT DIFFERENT SOUNDS.

22) WRITE A PROGRAM TO CREATE A CLASS CREDIT_CARD WITH ATTRIBUTES FOR CARD NUMBER, HOLDER NAME, AND BALANCE, AND METHODS TO PROCESS CHARGES AND PAYMENTS.

23) WRITE A PROGRAM TO IMPLEMENT AN INTERFACE PAYMENT WITH METHODS MAKE_PAYMENT() AND REFUND(), AND IMPLEMENT IT IN CLASSES CREDIT_CARD AND BANK_TRANSFER.

24) WRITE A PROGRAM TO DEMONSTRATE CONSTRUCTOR OVERLOADING IN A CLASS TIME, ALLOWING OBJECTS TO BE CREATED WITH DIFFERENT TIME FORMATS (E.G., HH , TOTAL MINUTES).

25) WRITE A PROGRAM TO CREATE A CLASS VECTOR WITH METHODS FOR VECTOR ADDITION, SUBTRACTION, AND SCALAR MULTIPLICATION.

26) WRITE A PROGRAM TO CREATE A CLASS E-COMMERCE_SYSTEM WITH SUBCLASSES USER, PRODUCT, AND ORDER. INCLUDE METHODS FOR USERS TO BROWSE PRODUCTS, PLACE ORDERS, AND VIEW ORDER HISTORY.

27) WRITE A PROGRAM TO IMPLEMENT AN ATM SIMULATION USING CLASSES ACCOUNT, ATM, AND TRANSACTION. INCLUDE METHODS FOR WITHDRAWING, DEPOSITING, AND CHECKING BALANCE WITHIN THE ATM CLASS.

28) WRITE A PROGRAM TO CREATE A CLASS FILE_MANAGER WITH METHODS TO READ AND WRITE TEXT FILES. IMPLEMENT EXCEPTION HANDLING TO CATCH FILE ACCESS ERRORS.

29) WRITE A PROGRAM TO IMPLEMENT AN INTERFACE SHAPE WITH A METHOD CALCULATE_AREA(). CREATE CLASSES CIRCLE, RECTANGLE, AND TRIANGLE THAT IMPLEMENT THIS INTERFACE.

30) WRITE A PROGRAM TO CREATE A CLASS STUDENT_DATABASE WITH METHODS TO ADD, DELETE, AND VIEW STUDENT RECORDS, USING A FILE TO PERSIST STUDENT DATA.

31) WRITE A PROGRAM TO CREATE A CLASS BANK_ACCOUNT WITH METHODS TO DEPOSIT AND WITHDRAW MONEY, AND CALCULATE INTEREST BASED ON ACCOUNT TYPE.

32) WRITE A PROGRAM TO CREATE A CLASS SHOPPING_CART WITH METHODS TO ADD, REMOVE, AND DISPLAY ITEMS IN THE CART, AND CALCULATE THE TOTAL COST.

33) WRITE A PROGRAM TO CREATE A CLASS LIBRARY_BOOK WITH ATTRIBUTES FOR TITLE, AUTHOR, AND ISBN. INCLUDE METHODS TO CHECK AVAILABILITY AND BORROW A BOOK.

34) WRITE A PROGRAM TO IMPLEMENT A CLASS QUEUE WITH METHODS TO ENQUEUE, DEQUEUE, AND DISPLAY THE ELEMENTS IN THE QUEUE.

35) WRITE A PROGRAM TO IMPLEMENT A CLASS STACK WITH METHODS TO PUSH, POP, AND DISPLAY THE STACK CONTENTS.

36) WRITE A PROGRAM TO CREATE A CLASS VEHICLE WITH ATTRIBUTES FOR MAKE, MODEL, AND YEAR. INCLUDE METHODS TO DISPLAY VEHICLE DETAILS.

37) WRITE A PROGRAM TO CREATE A CLASS PERSON WITH ATTRIBUTES FOR NAME, AGE, AND ADDRESS. INCLUDE A METHOD TO DISPLAY PERSONAL DETAILS.

38) WRITE A PROGRAM TO CREATE A CLASS SHOPPING_ITEM WITH ATTRIBUTES FOR ITEM NAME, PRICE, AND QUANTITY. INCLUDE A METHOD TO CALCULATE THE TOTAL COST.

39) WRITE A PROGRAM TO CREATE A CLASS CALENDAR WITH METHODS TO DISPLAY A SPECIFIED MONTH AND YEAR, AND TO DISPLAY EVENTS FOR A GIVEN DATE.

40) WRITE A PROGRAM TO CREATE A CLASS AIRLINE_BOOKING WITH METHODS TO BOOK A FLIGHT, CANCEL A FLIGHT, AND DISPLAY FLIGHT DETAILS.

41) WRITE A PROGRAM TO CREATE A CLASS WEATHER_REPORT WITH ATTRIBUTES FOR TEMPERATURE, HUMIDITY, AND WIND SPEED, AND A METHOD TO DISPLAY THE WEATHER.

42) WRITE A PROGRAM TO CREATE A CLASS STUDENT_GRADEBOOK WITH METHODS TO ADD STUDENT GRADES, CALCULATE AVERAGE, AND DISPLAY FINAL GRADES.

43) WRITE A PROGRAM TO IMPLEMENT A CLASS PERSONNEL_SYSTEM WITH CLASSES FOR EMPLOYEE, MANAGER, AND INTERN. INCLUDE METHODS TO DISPLAY EMPLOYEE DETAILS AND SALARY.

44) WRITE A PROGRAM TO CREATE A CLASS LIBRARY_MANAGEMENT_SYSTEM WITH METHODS TO ADD, REMOVE, AND SEARCH FOR BOOKS, AND TO DISPLAY ALL BOOKS.

45) WRITE A PROGRAM TO CREATE A CLASS INVENTORY_WITH_ITEM WITH ATTRIBUTES FOR ITEM NAME, ITEM CODE, AND QUANTITY. INCLUDE A METHOD TO CHECK STOCK LEVELS.

46) WRITE A PROGRAM TO CREATE A CLASS COMPUTER_SYSTEM WITH ATTRIBUTES FOR CPU, RAM, AND STORAGE, AND METHODS TO DISPLAY SYSTEM SPECIFICATIONS.

47) WRITE A PROGRAM TO CREATE A CLASS TEMPERATURE_SENSOR WITH METHODS TO RECORD TEMPERATURES, CALCULATE AVERAGE, AND DISPLAY SENSOR DATA.

48) WRITE A PROGRAM TO IMPLEMENT A CLASS RECIPE_BOOK WITH METHODS TO ADD, REMOVE, AND SEARCH FOR RECIPES, AND TO DISPLAY INGREDIENTS.

49) WRITE A PROGRAM TO CREATE A CLASS STUDENT_PROFILE WITH ATTRIBUTES FOR NAME, AGE, AND COURSE, AND A METHOD TO DISPLAY PROFILE DETAILS.

50) WRITE A PROGRAM TO IMPLEMENT A CLASS `TRACKER_WITH_TASKS` WITH METHODS TO ADD, REMOVE, AND LIST TASKS, AND TO MARK TASKS AS COMPLETED.

51) WRITE A PROGRAM TO CREATE A CLASS `BANK_TRANSACTION` WITH METHODS TO DEPOSIT, WITHDRAW, AND TRANSFER MONEY BETWEEN ACCOUNTS.

52) WRITE A PROGRAM TO CREATE A CLASS `HOTEL_RESERVATION` WITH METHODS TO MAKE A RESERVATION, CANCEL A RESERVATION, AND DISPLAY RESERVATION DETAILS.

53) WRITE A PROGRAM TO CREATE A CLASS `LIBRARY_MEMBER` WITH ATTRIBUTES FOR MEMBER NAME, ID, AND `BORROWED_BOOKS`, AND METHODS TO BORROW AND RETURN BOOKS.

54) WRITE A PROGRAM TO CREATE A CLASS SOCIAL_MEDIA_ACCOUNT WITH ATTRIBUTES FOR USERNAME, PASSWORD, AND PROFILE_PICTURE, AND METHODS TO DISPLAY PROFILE DETAILS.

55) WRITE A PROGRAM TO CREATE A CLASS BOOKSTORE WITH METHODS TO ADD, REMOVE, AND SEARCH FOR BOOKS, AND TO DISPLAY AVAILABLE STOCK.

56) WRITE A PROGRAM TO IMPLEMENT A CLASS COURSE_REGISTRATION WITH METHODS TO REGISTER FOR A COURSE, DROP A COURSE, AND DISPLAY COURSE DETAILS.

57) WRITE A PROGRAM TO CREATE A CLASS TICKET_BOOKING WITH METHODS TO BOOK A TICKET, CANCEL A TICKET, AND DISPLAY TICKET DETAILS.

58) WRITE A PROGRAM TO CREATE A CLASS CUSTOMER_ORDER WITH ATTRIBUTES FOR ORDER_ID, CUSTOMER_NAME, AND ORDER_STATUS, AND METHODS TO UPDATE STATUS.

59) WRITE A PROGRAM TO CREATE A CLASS GRADE_REPORT WITH METHODS TO ADD, REMOVE, AND UPDATE STUDENT GRADES, AND CALCULATE GRADE AVERAGES.

60) WRITE A PROGRAM TO CREATE A CLASS VEHICLE_RENTAL_SYSTEM WITH METHODS TO RENT VEHICLES, RETURN VEHICLES, AND DISPLAY VEHICLE DETAILS.

61) WRITE A PROGRAM TO CREATE A CLASS MOVIE_COLLECTION WITH METHODS TO ADD, REMOVE, AND SEARCH FOR MOVIES, AND TO DISPLAY MOVIE DETAILS.

62) WRITE A PROGRAM TO CREATE A CLASS JOB_APPLICATION WITH ATTRIBUTES FOR CANDIDATE_NAME, JOB_POSITION, AND APPLICATION_STATUS, AND METHODS TO UPDATE STATUS.

63) WRITE A PROGRAM TO CREATE A CLASS LIBRARY_CATALOG WITH METHODS TO ADD, REMOVE, AND SEARCH BOOKS, AND TO DISPLAY BOOK CATEGORIES.

64) WRITE A PROGRAM TO IMPLEMENT A CLASS GROCERY_LIST WITH METHODS TO ADD, REMOVE, AND DISPLAY ITEMS IN THE GROCERY LIST, AND CALCULATE TOTAL COST.

65) WRITE A PROGRAM TO CREATE A CLASS MUSIC_ALBUM WITH ATTRIBUTES FOR ARTIST, ALBUM_NAME, AND RELEASE_YEAR, AND METHODS TO DISPLAY ALBUM DETAILS.

66) WRITE A PROGRAM TO CREATE A CLASS ELECTRONIC_DEVICE WITH ATTRIBUTES FOR DEVICE_NAME, BRAND, AND PRICE, AND METHODS TO DISPLAY DEVICE SPECIFICATIONS.

67) WRITE A PROGRAM TO IMPLEMENT A CLASS FITNESS_TRACKER WITH METHODS TO RECORD WORKOUTS, TRACK CALORIES, AND DISPLAY FITNESS DATA.

68) WRITE A PROGRAM TO CREATE A CLASS BOOKING_SYSTEM WITH METHODS TO BOOK, CANCEL, AND MODIFY RESERVATIONS, AND DISPLAY RESERVATION DETAILS.

69) WRITE A PROGRAM TO CREATE A CLASS PRODUCT_INVENTORY WITH METHODS TO ADD, REMOVE, AND SEARCH FOR PRODUCTS, AND CALCULATE STOCK LEVELS.

70) WRITE A PROGRAM TO CREATE A CLASS PARKING_SYSTEM WITH METHODS TO PARK, UNPARK, AND DISPLAY PARKING SLOT DETAILS.

71) WRITE A PROGRAM TO CREATE A CLASS SCHOOL_COURSE WITH ATTRIBUTES FOR COURSE_NAME, COURSE_CODE, AND INSTRUCTOR, AND METHODS TO DISPLAY COURSE INFORMATION.

72) WRITE A PROGRAM TO CREATE A CLASS MOVIE_TICKET WITH ATTRIBUTES FOR MOVIE_NAME, SHOW_TIME, AND SEAT_NUMBER, AND METHODS TO DISPLAY TICKET DETAILS.

73) WRITE A PROGRAM TO CREATE A CLASS SHOPPING_LIST WITH ATTRIBUTES FOR ITEM_NAME, QUANTITY, AND PRICE, AND METHODS TO CALCULATE TOTAL COST.

74) WRITE A PROGRAM TO CREATE A CLASS MUSIC_PLAYER WITH METHODS TO PLAY, PAUSE, AND STOP MUSIC, AND TO DISPLAY SONG DETAILS.

75) WRITE A PROGRAM TO CREATE A CLASS RESTAURANT_ORDER WITH ATTRIBUTES FOR ORDER_ID, ORDER_DATE, AND ORDER_ITEMS, AND METHODS TO CALCULATE TOTAL COST.

76) WRITE A PROGRAM TO CREATE A CLASS WEATHER_STATION WITH METHODS TO RECORD TEMPERATURE, HUMIDITY, AND WIND_SPEED, AND DISPLAY WEATHER REPORTS.

77) WRITE A PROGRAM TO CREATE A CLASS TASK_MANAGER WITH METHODS TO ADD, REMOVE, AND UPDATE TASKS, AND DISPLAY TASKS BASED ON PRIORITY.

78) WRITE A PROGRAM TO CREATE A CLASS RENTAL_PROPERTY WITH ATTRIBUTES FOR PROPERTY_ID, LOCATION, AND RENTAL_PRICE, AND METHODS TO DISPLAY PROPERTY DETAILS.

79) WRITE A PROGRAM TO CREATE A CLASS BANK_ACCOUNT WITH ATTRIBUTES FOR ACCOUNT_NUMBER, ACCOUNT_BALANCE, AND ACCOUNT_TYPE, AND METHODS TO DEPOSIT AND WITHDRAW FUNDS.

80) WRITE A PROGRAM TO IMPLEMENT A CLASS TOLL_BOOTH WITH METHODS TO CALCULATE TOLL_FEES BASED ON VEHICLE_TYPE AND CALCULATE TOTAL COLLECTED FEES.

81) WRITE A PROGRAM TO CREATE A CLASS STUDENT_PROFILE WITH ATTRIBUTES FOR STUDENT_NAME, STUDENT_ID, AND COURSES_ENROLLED, AND METHODS TO DISPLAY PROFILE DETAILS.

82) WRITE A PROGRAM TO CREATE A CLASS SHIPPING_SERVICE WITH METHODS TO CALCULATE SHIPPING_COST BASED ON WEIGHT AND DESTINATION, AND TRACK SHIPMENT STATUS.

83) WRITE A PROGRAM TO CREATE A CLASS CAR_RENTAL_WITH ATTRIBUTES FOR RENTAL_DURATION, CAR_TYPE, AND COST, AND METHODS TO CALCULATE RENTAL COST.

84) WRITE A PROGRAM TO CREATE A CLASS PATIENT_RECORD WITH ATTRIBUTES FOR PATIENT_NAME, AGE, AND MEDICAL_HISTORY, AND METHODS TO UPDATE AND DISPLAY RECORDS.

85) WRITE A PROGRAM TO CREATE A CLASS EMPLOYEE_PAYROLL WITH ATTRIBUTES FOR EMPLOYEE_NAME, DESIGNATION, AND SALARY, AND METHODS TO CALCULATE PAYMENTS.

86) WRITE A PROGRAM TO CREATE A CLASS GAME_LEVEL WITH ATTRIBUTES FOR LEVEL_NUMBER, LEVEL_NAME, AND DIFFICULTY, AND METHODS TO DISPLAY LEVEL DETAILS.

87) WRITE A PROGRAM TO CREATE A CLASS STOCK_PORTFOLIO WITH ATTRIBUTES FOR STOCK_SYMBOL, PURCHASE_PRICE, AND QUANTITY, AND METHODS TO DISPLAY PORTFOLIO DETAILS.

88) WRITE A PROGRAM TO CREATE A CLASS DIGITAL_LIBRARY WITH METHODS TO ADD, REMOVE, AND SEARCH FOR EBOOKS, AND DISPLAY LIBRARY CONTENTS.

89) WRITE A PROGRAM TO CREATE A CLASS HOTEL_BOOKING WITH ATTRIBUTES FOR HOTEL_NAME, ROOM_TYPE, AND STAY_DURATION, AND METHODS TO CALCULATE TOTAL COST.

90) WRITE A PROGRAM TO CREATE A CLASS CUSTOMER_REVIEW WITH ATTRIBUTES FOR CUSTOMER_NAME, REVIEW_RATING, AND REVIEW_COMMENTS, AND METHODS TO DISPLAY REVIEWS.

91) WRITE A PROGRAM TO CREATE A CLASS TRAVEL_AGENCY WITH ATTRIBUTES FOR PACKAGE_NAME, DESTINATION, AND COST, AND METHODS TO DISPLAY TRAVEL PACKAGE DETAILS.

92) WRITE A PROGRAM TO CREATE A CLASS MUSIC_PLAYLIST WITH METHODS TO ADD, REMOVE, AND REORDER SONGS, AND DISPLAY PLAYLIST DETAILS.

93) WRITE A PROGRAM TO CREATE A CLASS PAYMENT_GATEWAY WITH METHODS TO PROCESS_PAYMENTS, REFUND, AND VERIFY_TRANSACTION, AND DISPLAY PAYMENT STATUS.

94) WRITE A PROGRAM TO CREATE A CLASS SCHOOL_TIMETABLE WITH METHODS TO ADD, REMOVE, AND UPDATE CLASSES, AND DISPLAY TIMETABLE.

95) WRITE A PROGRAM TO CREATE A CLASS `FITNESS_APP` WITH METHODS TO LOG WORKOUTS, TRACK PROGRESS, AND DISPLAY FITNESS REPORTS.

96) WRITE A PROGRAM TO CREATE A CLASS `VEHICLE_MAINTENANCE` WITH ATTRIBUTES FOR `VEHICLE_MODEL`, `SERVICE_DATE`, AND `SERVICE_TYPE`, AND METHODS TO DISPLAY MAINTENANCE DETAILS.

97) WRITE A PROGRAM TO CREATE A CLASS `AIRLINE_RESERVATION` WITH METHODS TO `BOOK_FLIGHT`, `CANCEL_FLIGHT`, AND DISPLAY FLIGHT DETAILS.

98) WRITE A PROGRAM TO CREATE A CLASS `LOAN_CALCULATOR` WITH METHODS TO `CALCULATE` `LOAN_AMOUNT`, `INTEREST_RATE`, AND `LOAN_TERM`.

99) WRITE A PROGRAM TO CREATE A CLASS `ONLINE_STORE` WITH METHODS TO `ADD_PRODUCT`, `REMOVE_PRODUCT`, AND `DISPLAY_PRODUCT_DETAILS`.

100) WRITE A PROGRAM TO CREATE A CLASS EVENT_MANAGEMENT WITH METHODS TO SCHEDULE_EVENT, CANCEL_EVENT, AND DISPLAY EVENT DETAILS.

=> 13. FILE HANDLING

1) WRITE A PROGRAM TO CREATE A TEXT FILE AND WRITE THE FIRST TEN MULTIPLES OF A NUMBER INPUTTED BY THE USER.

2) WRITE A PROGRAM TO READ THE CONTENTS OF A TEXT FILE LINE BY LINE AND DISPLAY EACH LINE ON THE SCREEN.

3) WRITE A PROGRAM TO COUNT AND DISPLAY THE NUMBER OF LINES, WORDS, AND CHARACTERS IN A GIVEN TEXT FILE.

4) WRITE A PROGRAM TO COPY CONTENT FROM ONE TEXT FILE TO ANOTHER FILE.

5) WRITE A PROGRAM TO APPEND USER-ENTERED TEXT TO AN EXISTING FILE AND THEN DISPLAY THE UPDATED FILE CONTENTS.

6) WRITE A PROGRAM TO READ A FILE AND DISPLAY ONLY THOSE LINES THAT CONTAIN A SPECIFIC WORD INPUTTED BY THE USER.

7) WRITE A PROGRAM TO MERGE TWO FILES INTO A THIRD FILE, WHERE THE CONTENTS OF BOTH FILES ARE COMBINED LINE BY LINE.

8) WRITE A PROGRAM TO SORT THE CONTENT OF A TEXT FILE ALPHABETICALLY BY EACH LINE AND SAVE IT INTO A NEW FILE.

9) WRITE A PROGRAM TO SEARCH FOR A GIVEN WORD IN A FILE AND DISPLAY THE LINE NUMBERS WHERE THE WORD OCCURS.

10) WRITE A PROGRAM TO REMOVE ALL WHITESPACE CHARACTERS (E.G., SPACES, TABS, NEWLINES) FROM A TEXT FILE AND SAVE THE RESULT INTO A NEW FILE.

11) WRITE A PROGRAM TO CREATE A LOG FILE THAT RECORDS EVERY TIME A USER RUNS THE PROGRAM, INCLUDING THE DATE AND TIME OF EXECUTION.

12) WRITE A PROGRAM TO CREATE A STUDENT DATABASE IN A FILE WHERE EACH STUDENT'S DETAILS ARE SAVED AS A NEW LINE. ALLOW THE USER TO ADD, UPDATE, AND DELETE STUDENT RECORDS.

13) WRITE A PROGRAM TO IMPLEMENT A SIMPLE PASSWORD MANAGER, STORING ENCRYPTED USERNAME-PASSWORD PAIRS IN A FILE. INCLUDE OPTIONS TO ADD, VIEW, AND DELETE PASSWORD ENTRIES.

14) WRITE A PROGRAM TO CREATE A CSV FILE AND STORE STUDENT SCORES IN IT. THEN, READ THE FILE AND CALCULATE THE AVERAGE SCORE FOR EACH STUDENT.

15) WRITE A PROGRAM TO READ A TEXT FILE AND FIND THE FREQUENCY OF EACH WORD IN THE FILE. STORE THE FREQUENCIES IN A NEW FILE IN DESCENDING ORDER.

=> 14. EXCEPTION HANDLING

1) WRITE A PROGRAM TO HANDLE DIVISION BY ZERO EXCEPTION WHILE PERFORMING DIVISION.

2) WRITE A PROGRAM TO HANDLE INVALID INDEX EXCEPTION WHILE ACCESSING ELEMENTS IN AN ARRAY.

3) WRITE A PROGRAM TO HANDLE NULL POINTER EXCEPTION WHEN ACCESSING UNINITIALIZED OBJECTS.

4) WRITE A PROGRAM TO HANDLE FILE NOT FOUND EXCEPTION WHEN OPENING A NON-EXISTENT FILE.

5) WRITE A PROGRAM TO HANDLE NUMBER FORMAT EXCEPTION WHEN CONVERTING STRING TO INTEGER.

6) WRITE A PROGRAM TO HANDLE ARRAY INDEX OUT OF BOUNDS EXCEPTION DURING ARRAY OPERATIONS.

7) WRITE A PROGRAM TO HANDLE STRING INDEX OUT OF BOUNDS EXCEPTION WHILE MANIPULATING STRINGS.

8) WRITE A PROGRAM TO HANDLE ILLEGAL ARGUMENT EXCEPTION IN A METHOD.

9) WRITE A PROGRAM TO CATCH AND HANDLE MULTIPLE EXCEPTIONS USING A SINGLE CATCH BLOCK.

10) WRITE A PROGRAM TO HANDLE ARITHMETIC EXCEPTION WHILE PERFORMING MODULUS OPERATION.

11) WRITE A PROGRAM TO CREATE A CUSTOM EXCEPTION FOR INVALID AGE INPUT AND HANDLE IT.

12) WRITE A PROGRAM TO HANDLE CLASS NOT FOUND EXCEPTION WHEN LOADING A CLASS DYNAMICALLY.

13) WRITE A PROGRAM TO HANDLE IO EXCEPTION WHILE READING FROM A FILE.

14) WRITE A PROGRAM TO HANDLE NO SUCH ELEMENT EXCEPTION WHEN ACCESSING A MISSING ELEMENT IN A COLLECTION.

15) WRITE A PROGRAM TO HANDLE ILLEGAL STATE EXCEPTION WHILE MODIFYING A LIST DURING ITERATION.

16) WRITE A PROGRAM TO HANDLE ILLEGAL ACCESS EXCEPTION WHEN ACCESSING A PRIVATE FIELD USING REFLECTION.

17) WRITE A PROGRAM TO HANDLE UNSUPPORTED OPERATION EXCEPTION WHEN REMOVING ELEMENT FROM AN UNMODIFIABLE COLLECTION.

18) WRITE A PROGRAM TO HANDLE CLASS CAST EXCEPTION WHEN CASTING OBJECTS INCORRECTLY.

19) WRITE A PROGRAM TO HANDLE SQL EXCEPTION WHILE EXECUTING A DATABASE QUERY.

20) WRITE A PROGRAM TO HANDLE MALFORMED URL EXCEPTION WHEN PARSING AN INVALID URL.

21) WRITE A PROGRAM TO HANDLE INTERRUPTED EXCEPTION WHEN A THREAD IS INTERRUPTED DURING SLEEP.

22) WRITE A PROGRAM TO HANDLE SECURITY EXCEPTION WHEN ACCESSING A RESTRICTED RESOURCE.

23) WRITE A PROGRAM TO CATCH AND HANDLE EXCEPTIONS THROWN BY MULTIPLE NESTED METHODS.

24) WRITE A PROGRAM TO HANDLE EMPTY STACK EXCEPTION WHEN POPPING FROM AN EMPTY STACK.

25) WRITE A PROGRAM TO HANDLE ILLEGAL MONITOR STATE EXCEPTION WHEN ACCESSING SYNCHRONIZED BLOCKS INCORRECTLY.

26) WRITE A PROGRAM TO HANDLE INVALID CHARACTER EXCEPTION WHEN READING FROM A FILE.

27) WRITE A PROGRAM TO HANDLE ILLEGAL THREAD STATE EXCEPTION WHEN STARTING A THREAD MULTIPLE TIMES.

28) WRITE A PROGRAM TO HANDLE INSTANTIATION EXCEPTION WHILE CREATING AN INSTANCE OF AN ABSTRACT CLASS.

29) WRITE A PROGRAM TO HANDLE CLONE NOT SUPPORTED EXCEPTION WHILE CLONING AN OBJECT WITHOUT IMPLEMENTING CLONEABLE.

30) WRITE A PROGRAM TO HANDLE ARRAY STORE EXCEPTION WHEN ASSIGNING INCOMPATIBLE TYPES IN AN ARRAY.

31) WRITE A PROGRAM TO HANDLE PARSE EXCEPTION WHEN PARSING AN INCORRECT DATE FORMAT.

32) WRITE A PROGRAM TO HANDLE MISSING RESOURCE EXCEPTION WHEN ACCESSING A MISSING RESOURCE BUNDLE.

33) WRITE A PROGRAM TO HANDLE BUFFER OVERFLOW EXCEPTION WHEN WRITING TO A FULL BUFFER.

34) WRITE A PROGRAM TO HANDLE BUFFER UNDERFLOW EXCEPTION WHEN READING FROM AN EMPTY BUFFER.

35) WRITE A PROGRAM TO HANDLE TIMEOUT EXCEPTION WHEN A REQUEST TAKES TOO LONG TO COMPLETE.

36) WRITE A PROGRAM TO HANDLE CONNECT EXCEPTION WHEN UNABLE TO ESTABLISH A NETWORK CONNECTION.

37) WRITE A PROGRAM TO HANDLE SOCKET EXCEPTION WHEN PERFORMING SOCKET OPERATIONS.

38) WRITE A PROGRAM TO HANDLE EOF EXCEPTION WHEN READING PAST THE END OF A FILE.

39) WRITE A PROGRAM TO HANDLE INVALID PATH EXCEPTION WHEN ACCESSING A FILE AT AN INVALID PATH.

40) WRITE A PROGRAM TO HANDLE CHARACTER CODING EXCEPTION WHEN ENCODING OR DECODING TEXT.

41) WRITE A PROGRAM TO HANDLE ILLEGAL CHANNEL GROUP EXCEPTION WHILE PERFORMING OPERATIONS ON A CHANNEL GROUP.

42) WRITE A PROGRAM TO HANDLE DIRECTORY NOT EMPTY EXCEPTION WHILE DELETING A DIRECTORY WITH CONTENTS.

43) WRITE A PROGRAM TO HANDLE FILE SYSTEM LOOP EXCEPTION WHEN LINKING FILES IN A CIRCULAR MANNER.

44) WRITE A PROGRAM TO HANDLE ACCESS DENIED EXCEPTION WHEN TRYING TO MODIFY A READ-ONLY FILE.

45) WRITE A PROGRAM TO HANDLE ILLEGAL BLOCK SIZE EXCEPTION DURING CRYPTOGRAPHIC OPERATIONS.

46) WRITE A PROGRAM TO HANDLE KEY EXCEPTION WHEN ACCESSING AN INVALID KEY IN A MAP.

47) WRITE A PROGRAM TO HANDLE IO EXCEPTION WHEN COPYING LARGE FILES.

48) WRITE A PROGRAM TO HANDLE EXCESSIVE RETRIES EXCEPTION WHEN A TASK EXCEEDS RETRY LIMIT.

49) WRITE A PROGRAM TO HANDLE OUT OF MEMORY EXCEPTION WHEN CREATING LARGE ARRAYS.

50) WRITE A PROGRAM TO HANDLE CONCURRENT MODIFICATION EXCEPTION WHEN MODIFYING A COLLECTION IN MULTITHREADED ENVIRONMENT.

51) WRITE A PROGRAM TO HANDLE CIRCULAR DEPENDENCY EXCEPTION WHEN INITIALIZING OBJECTS WITH CIRCULAR DEPENDENCIES.

52) WRITE A PROGRAM TO HANDLE ENUM CONSTANT NOT PRESENT EXCEPTION WHEN ACCESSING AN UNDEFINED ENUM CONSTANT.

53) WRITE A PROGRAM TO HANDLE ILLEGAL FORMAT EXCEPTION WHEN PRINTING WITH INCORRECT FORMAT SPECIFIERS.

54) WRITE A PROGRAM TO HANDLE READ ONLY BUFFER EXCEPTION WHEN MODIFYING A READ-ONLY BUFFER.

55) WRITE A PROGRAM TO HANDLE STACK OVERFLOW EXCEPTION WHEN CALLING A METHOD RECURSIVELY WITHOUT BASE CASE.

56) WRITE A PROGRAM TO HANDLE DATE TIME EXCEPTION WHEN PARSING AN INVALID DATE OR TIME.

57) WRITE A PROGRAM TO HANDLE NUMBER FORMAT EXCEPTION WHEN PARSING NON-NUMERIC STRING TO NUMBER.

58) WRITE A PROGRAM TO HANDLE ILLEGAL ACCESS EXCEPTION WHEN INVOKING A PRIVATE METHOD VIA REFLECTION.

59) WRITE A PROGRAM TO HANDLE MISSING FORMAT ARGUMENT EXCEPTION WHEN PRINTING STRING WITHOUT ENOUGH ARGUMENTS.

60) WRITE A PROGRAM TO HANDLE SECURITY EXCEPTION WHEN ACCESSING A RESTRICTED CLASS OR METHOD.

61) WRITE A PROGRAM TO HANDLE UNCHECKED EXCEPTION AND DEMONSTRATE HOW IT DIFFERS FROM CHECKED EXCEPTIONS.

62) WRITE A PROGRAM TO HANDLE TOO MANY FILE OPEN EXCEPTION WHEN OPENING EXCESSIVE FILES WITHOUT CLOSING.

63) WRITE A PROGRAM TO HANDLE EMPTY STACK EXCEPTION WHILE EVALUATING EXPRESSIONS USING STACK.

64) WRITE A PROGRAM TO HANDLE IO EXCEPTION WHEN ACCESSING NON-EXISTENT HARDWARE DEVICE.

65) WRITE A PROGRAM TO HANDLE ILLEGAL THREAD STATE EXCEPTION WHILE PERFORMING ILLEGAL THREAD OPERATIONS.

=> 15. THREADING

1) WRITE A PROGRAM TO CREATE AND START A SIMPLE THREAD THAT PRINTS "HELLO FROM THREAD".

2) WRITE A PROGRAM TO CREATE MULTIPLE THREADS AND PRINT THEIR NAMES.

3) WRITE A PROGRAM TO CREATE A THREAD THAT CALCULATES THE SUM OF AN ARRAY OF NUMBERS.

4) WRITE A PROGRAM TO USE MULTIPLE THREADS TO COMPUTE FACTORIAL OF DIFFERENT NUMBERS SIMULTANEOUSLY.

5) WRITE A PROGRAM TO IMPLEMENT A TIMER USING A SEPARATE THREAD TO COUNT SECONDS.

6) WRITE A PROGRAM TO DEMONSTRATE SYNCHRONIZATION BY HAVING MULTIPLE THREADS INCREMENT A SHARED VARIABLE.

7) WRITE A PROGRAM TO CALCULATE THE SUM OF TWO ARRAYS CONCURRENTLY USING TWO THREADS.

8) WRITE A PROGRAM TO CREATE A THREAD THAT PRINTS EVEN NUMBERS AND ANOTHER THREAD THAT PRINTS ODD NUMBERS.

9) WRITE A PROGRAM TO CREATE A THREAD THAT PERFORMS MATRIX MULTIPLICATION WHILE THE MAIN THREAD DOES ANOTHER TASK.

10) WRITE A PROGRAM TO DEMONSTRATE THREAD PRIORITY BY CREATING HIGH AND LOW PRIORITY THREADS.

11) WRITE A PROGRAM TO SIMULATE A SIMPLE BANK ACCOUNT WITH DEPOSIT AND WITHDRAW FUNCTIONS USING SYNCHRONIZED METHODS.

12) WRITE A PROGRAM TO CREATE A THREAD THAT CALCULATES THE FIBONACCI SERIES UP TO A GIVEN NUMBER.

13) WRITE A PROGRAM TO DEMONSTRATE THREAD INTERRUPTION BY CREATING A THREAD THAT SLEEPS AND IS INTERRUPTED BY MAIN THREAD.

14) WRITE A PROGRAM TO IMPLEMENT A PRODUCER-CONSUMER PROBLEM USING THREADS.

15) WRITE A PROGRAM TO IMPLEMENT A SIMPLE COUNTER USING MULTIPLE THREADS WITH LOCKING.

16) WRITE A PROGRAM TO IMPLEMENT THREAD JOINING WHERE ONE THREAD WAITS FOR ANOTHER TO COMPLETE.

17) WRITE A PROGRAM TO CREATE A THREAD POOL THAT EXECUTES A LIST OF TASKS SIMULTANEOUSLY.

18) WRITE A PROGRAM TO PRINT CURRENT THREAD ID AND NAME FOR MULTIPLE THREADS.

19) WRITE A PROGRAM TO IMPLEMENT A READER-WRITER PROBLEM USING MULTIPLE THREADS.

20) WRITE A PROGRAM TO MEASURE THREAD EXECUTION TIME FOR A SET OF COMPUTATIONS.

21) WRITE A PROGRAM TO IMPLEMENT A DEADLOCK SCENARIO USING TWO THREADS AND TWO RESOURCES.

22) WRITE A PROGRAM TO DEMONSTRATE THREAD GROUPS BY CREATING MULTIPLE THREADS IN A GROUP AND DISPLAYING GROUP DETAILS.

23) WRITE A PROGRAM TO SIMULATE AN ONLINE TICKET BOOKING SYSTEM WITH MULTIPLE THREADS ACCESSING SEATS.

24) WRITE A PROGRAM TO DEMONSTRATE RACE CONDITION AND HOW TO PREVENT IT USING SYNCHRONIZATION.

25) WRITE A PROGRAM TO DEMONSTRATE THE USE OF VOLATILE VARIABLE IN THREADING.

26) WRITE A PROGRAM TO CREATE A THREAD THAT PRINTS A COUNTDOWN FROM A SPECIFIED NUMBER TO ZERO.

27) WRITE A PROGRAM TO SIMULATE A SIMPLE CHAT SYSTEM BETWEEN TWO THREADS.

28) WRITE A PROGRAM TO PERFORM MATRIX ADDITION CONCURRENTLY USING MULTIPLE THREADS.

29) WRITE A PROGRAM TO SORT AN ARRAY USING MULTIPLE THREADS FOR DIVIDED SECTIONS OF THE ARRAY.

30) WRITE A PROGRAM TO SIMULATE THE DINING PHILOSOPHERS PROBLEM USING MULTIPLE THREADS.

31) WRITE A PROGRAM TO IMPLEMENT A THREAD THAT CONTINUOUSLY CHECKS IF A FILE HAS BEEN MODIFIED.

32) WRITE A PROGRAM TO DEMONSTRATE INTER-THREAD COMMUNICATION USING WAIT AND NOTIFY.

33) WRITE A PROGRAM TO SIMULATE A REAL-TIME CLOCK USING A THREAD THAT UPDATES EVERY SECOND.

34) WRITE A PROGRAM TO CREATE AND START A DAEMON THREAD THAT RUNS IN THE BACKGROUND.

35) WRITE A PROGRAM TO DEMONSTRATE THREAD LOCAL VARIABLES BY STORING UNIQUE VALUES PER THREAD.

36) WRITE A PROGRAM TO MANAGE MULTIPLE THREADS THAT CALCULATE THE FACTORIAL OF NUMBERS SIMULTANEOUSLY.

37) WRITE A PROGRAM TO CALCULATE ELEMENT-WISE MULTIPLICATION OF TWO ARRAYS USING MULTIPLE THREADS.

38) WRITE A PROGRAM TO IMPLEMENT AN ELEVATOR SIMULATION USING THREADS TO MOVE BETWEEN FLOORS.

39) WRITE A PROGRAM TO CREATE A THREAD THAT COMPUTES THE SUM OF EVEN AND ANOTHER THREAD FOR ODD NUMBERS IN A LIST.

40) WRITE A PROGRAM TO SIMULATE FILE DOWNLOADS USING MULTIPLE THREADS FOR DIFFERENT PARTS OF A FILE.

41) WRITE A PROGRAM TO IMPLEMENT A SIMPLE THREAD SAFE QUEUE USING SYNCHRONIZED METHODS.

42) WRITE A PROGRAM TO CALCULATE THE PRIME NUMBERS IN A GIVEN RANGE USING MULTIPLE THREADS.

43) WRITE A PROGRAM TO CONTROL THE SPEED OF A THREAD BY MODIFYING ITS SLEEP TIME.

44) WRITE A PROGRAM TO IMPLEMENT A LOGGING SYSTEM WHERE MULTIPLE THREADS LOG EVENTS SIMULTANEOUSLY.

45) WRITE A PROGRAM TO SIMULATE AN ASYNCHRONOUS TASK SYSTEM WHERE TASKS EXECUTE INDEPENDENTLY USING THREADS.

46) WRITE A PROGRAM TO SIMULATE A PRINT QUEUE SYSTEM WITH MULTIPLE THREADS REPRESENTING PRINTERS.

47) WRITE A PROGRAM TO IMPLEMENT A COUNTDOWN TIMER WITH THREADS DISPLAYING TIME LEFT EVERY SECOND.

48) WRITE A PROGRAM TO CREATE THREADS THAT EXECUTE A BATCH OF TASKS IN PARALLEL USING EXECUTOR SERVICE.

49) WRITE A PROGRAM TO MEASURE CPU UTILIZATION BY CREATING MULTIPLE COMPUTATION-INTENSIVE THREADS.

50) WRITE A PROGRAM TO CREATE A THREAD THAT CALCULATES FACTORIAL OF LARGE NUMBERS USING BIGINTEGER.

51) WRITE A PROGRAM TO IMPLEMENT A SEMAPHORE TO LIMIT ACCESS TO A RESOURCE USED BY MULTIPLE THREADS.

52) WRITE A PROGRAM TO IMPLEMENT A BARRIER THAT WAITS FOR A SET NUMBER OF THREADS TO COMPLETE.

53) WRITE A PROGRAM TO SIMULATE CUSTOMER SERVICE REPRESENTATIVES ATTENDING CALLS USING MULTIPLE THREADS.

54) WRITE A PROGRAM TO SIMULATE A TIMER THAT EXECUTES MULTIPLE TASKS AT INTERVALS USING MULTIPLE THREADS.

55) WRITE A PROGRAM TO SIMULATE A HOTEL BOOKING SYSTEM WHERE MULTIPLE THREADS BOOK ROOMS CONCURRENTLY.

56) WRITE A PROGRAM TO IMPLEMENT A TRAFFIC LIGHT SYSTEM USING MULTIPLE THREADS THAT MANAGE DIFFERENT LANES.

57) WRITE A PROGRAM TO EXECUTE A TASK IN PARALLEL USING FORK-JOIN POOL TO SPLIT AND MERGE RESULTS.

58) WRITE A PROGRAM TO SORT A LARGE ARRAY USING PARALLEL THREADS FOR EACH SUBARRAY.

59) WRITE A PROGRAM TO IMPLEMENT A SIMPLE COUNTDOWN LATCH THAT WAITS FOR MULTIPLE THREADS TO COMPLETE.

60) WRITE A PROGRAM TO SIMULATE A MULTI-USER BANK ACCOUNT SYSTEM WITH CONCURRENT ACCESS USING THREADS.

61) WRITE A PROGRAM TO SIMULATE A GROCERY STORE CHECKOUT WITH MULTIPLE CASHIER THREADS.

62) WRITE A PROGRAM TO IMPLEMENT A TIMEOUT SYSTEM WHERE A THREAD TERMINATES AFTER SPECIFIED TIME IF NOT COMPLETED.

63) WRITE A PROGRAM TO DEMONSTRATE ASYNCHRONOUS FILE READING USING MULTIPLE THREADS.

64) WRITE A PROGRAM TO IMPLEMENT A LOAD BALANCER USING THREADS THAT DISTRIBUTE TASKS TO WORKERS.

65) WRITE A PROGRAM TO CREATE MULTIPLE THREADS THAT PRINT NUMBERS FROM DIFFERENT RANGES IN PARALLEL.

=> 16. POINTERS

1) WRITE A PROGRAM TO DEMONSTRATE POINTER USAGE IN A SIMPLE CONTEXT (INITIALIZE AND PRINT).

2) WRITE A PROGRAM TO SWAP TWO NUMBERS USING POINTERS.

3) WRITE A PROGRAM TO FIND THE LENGTH OF A STRING USING POINTERS.

4) WRITE A PROGRAM TO COPY A STRING TO ANOTHER STRING USING POINTERS.

5) WRITE A PROGRAM TO REVERSE A STRING USING POINTERS.

6) WRITE A PROGRAM TO FIND THE LARGEST ELEMENT IN AN ARRAY USING POINTERS.

7) WRITE A PROGRAM TO SORT AN ARRAY OF INTEGERS USING POINTERS.

8) WRITE A PROGRAM TO ADD TWO MATRICES USING POINTERS.

9) WRITE A PROGRAM TO SUBTRACT TWO MATRICES USING POINTERS.

10) WRITE A PROGRAM TO FIND THE SUM OF ALL ELEMENTS IN AN ARRAY USING POINTERS.

11) WRITE A PROGRAM TO PERFORM BUBBLE SORT ON AN ARRAY USING POINTERS.

12) WRITE A PROGRAM TO FIND THE MINIMUM AND MAXIMUM ELEMENTS IN AN ARRAY USING POINTERS.

13) WRITE A PROGRAM TO IMPLEMENT LINEAR SEARCH USING POINTERS.

14) WRITE A PROGRAM TO IMPLEMENT BINARY SEARCH USING POINTERS.

15) WRITE A PROGRAM TO FIND THE FACTORIAL OF A NUMBER USING POINTERS.

16) WRITE A PROGRAM TO FIND THE FIBONACCI SERIES UP TO N ELEMENTS USING POINTERS.

17) WRITE A PROGRAM TO SWAP TWO STRINGS USING POINTERS.

18) WRITE A PROGRAM TO FIND THE DOT PRODUCT OF TWO VECTORS USING POINTERS.

19) WRITE A PROGRAM TO TRANSPOSE A MATRIX USING POINTERS.

20) WRITE A PROGRAM TO CHECK IF A GIVEN STRING IS A PALINDROME USING POINTERS.

21) WRITE A PROGRAM TO CONCATENATE TWO STRINGS USING POINTERS.

22) WRITE A PROGRAM TO MULTIPLY TWO MATRICES USING POINTERS.

23) WRITE A PROGRAM TO FIND THE AVERAGE OF ELEMENTS IN AN ARRAY USING POINTERS.

24) WRITE A PROGRAM TO ACCESS AND MODIFY ARRAY ELEMENTS USING POINTER ARITHMETIC.

25) WRITE A PROGRAM TO DEMONSTRATE POINTER TO POINTER CONCEPT.

26) WRITE A PROGRAM TO FIND THE LENGTH OF A LINKED LIST USING POINTERS.

27) WRITE A PROGRAM TO DELETE AN ELEMENT FROM AN ARRAY USING POINTERS.

28) WRITE A PROGRAM TO INSERT AN ELEMENT AT THE BEGINNING OF A LINKED LIST USING POINTERS.

29) WRITE A PROGRAM TO INSERT AN ELEMENT AT THE END OF A LINKED LIST USING POINTERS.

30) WRITE A PROGRAM TO REMOVE DUPLICATES FROM A LINKED LIST USING POINTERS.

31) WRITE A PROGRAM TO REVERSE A LINKED LIST USING POINTERS.

32) WRITE A PROGRAM TO CREATE A DOUBLE-LINKED LIST USING POINTERS.

33) WRITE A PROGRAM TO DELETE AN ELEMENT FROM A LINKED LIST USING POINTERS.

34) WRITE A PROGRAM TO IMPLEMENT STACK OPERATIONS USING POINTERS.

35) WRITE A PROGRAM TO IMPLEMENT QUEUE OPERATIONS USING POINTERS.

36) WRITE A PROGRAM TO COPY AN ARRAY INTO ANOTHER ARRAY USING POINTERS.

37) WRITE A PROGRAM TO FIND THE COMMON ELEMENTS BETWEEN TWO ARRAYS USING POINTERS.

38) WRITE A PROGRAM TO SORT A LINKED LIST USING POINTERS.

39) WRITE A PROGRAM TO SWAP NODES IN A LINKED LIST USING POINTERS.

40) WRITE A PROGRAM TO FIND THE SUM OF DIAGONAL ELEMENTS OF A MATRIX USING POINTERS.

41) WRITE A PROGRAM TO IMPLEMENT A CIRCULAR LINKED LIST USING POINTERS.

42) WRITE A PROGRAM TO FIND THE SIZE OF A STRUCTURE USING POINTERS.

43) WRITE A PROGRAM TO COPY THE CONTENT OF ONE STRUCTURE TO ANOTHER USING POINTERS.

44) WRITE A PROGRAM TO DEMONSTRATE VOID POINTER USAGE.

45) WRITE A PROGRAM TO SHOW FUNCTION POINTERS AND USE THEM IN SIMPLE OPERATIONS.

46) WRITE A PROGRAM TO IMPLEMENT DYNAMIC MEMORY ALLOCATION FOR AN ARRAY USING POINTERS.

47) WRITE A PROGRAM TO CONCATENATE TWO LINKED LISTS USING POINTERS.

48) WRITE A PROGRAM TO IMPLEMENT BINARY TREE OPERATIONS USING POINTERS.

49) WRITE A PROGRAM TO FIND THE DEPTH OF A BINARY TREE USING POINTERS.

50) WRITE A PROGRAM TO COUNT THE NUMBER OF LEAF NODES IN A BINARY TREE USING POINTERS.

51) WRITE A PROGRAM TO FIND THE HEIGHT OF A BINARY TREE USING POINTERS.

52) WRITE A PROGRAM TO DELETE A BINARY TREE USING POINTERS.

53) WRITE A PROGRAM TO IMPLEMENT A BINARY SEARCH TREE USING POINTERS.

54) WRITE A PROGRAM TO FIND THE INORDER SUCCESSOR OF A NODE IN A BST USING POINTERS.

55) WRITE A PROGRAM TO DISPLAY THE MEMORY ADDRESS OF EACH ELEMENT IN AN ARRAY USING POINTERS.

56) WRITE A PROGRAM TO FIND THE SUM OF DIGITS OF A NUMBER USING POINTERS.

57) WRITE A PROGRAM TO PERFORM MERGE SORT USING POINTERS.

58) WRITE A PROGRAM TO SORT STRINGS IN LEXICOGRAPHICAL ORDER USING POINTERS.

59) WRITE A PROGRAM TO REMOVE DUPLICATE ELEMENTS IN AN ARRAY USING POINTERS.

60) WRITE A PROGRAM TO CREATE A LINKED LIST USING DYNAMIC MEMORY ALLOCATION AND POINTERS.

61) WRITE A PROGRAM TO FIND THE MEDIAN OF AN ARRAY USING POINTERS.

62) WRITE A PROGRAM TO IMPLEMENT A DEQUE USING POINTERS.

63) WRITE A PROGRAM TO FLATTEN A MULTI-LEVEL LINKED LIST USING POINTERS.

64) WRITE A PROGRAM TO DEMONSTRATE POINTERS AND ARRAYS BY IMPLEMENTING A CHARACTER ARRAY.

65) WRITE A PROGRAM TO FIND ALL PRIME NUMBERS BETWEEN TWO INTEGERS USING POINTERS.

66) WRITE A PROGRAM TO SWAP VALUES OF VARIABLES USING POINTERS WITHOUT A THIRD VARIABLE.

67) WRITE A PROGRAM TO CONCATENATE MULTIPLE ARRAYS USING POINTERS.

68) WRITE A PROGRAM TO FIND THE NTH ELEMENT FROM THE END IN A LINKED LIST USING POINTERS.

69) WRITE A PROGRAM TO FIND THE MIDDLE ELEMENT OF A LINKED LIST USING POINTERS.

70) WRITE A PROGRAM TO IMPLEMENT HASHING WITH SEPARATE CHAINING USING POINTERS.

71) WRITE A PROGRAM TO CLONE A LINKED LIST WITH RANDOM POINTERS USING POINTERS.

72) WRITE A PROGRAM TO IMPLEMENT A MINIMUM SPANNING TREE USING POINTERS.

73) WRITE A PROGRAM TO FIND THE INTERSECTION POINT OF TWO LINKED LISTS USING POINTERS.

74) WRITE A PROGRAM TO IMPLEMENT A GRAPH USING ADJACENCY LISTS WITH POINTERS.

75) WRITE A PROGRAM TO FIND THE UNION OF TWO SETS USING POINTERS.

76) WRITE A PROGRAM TO IMPLEMENT A SPARSE MATRIX USING POINTERS.

77) WRITE A PROGRAM TO CHECK IF TWO LINKED LISTS ARE IDENTICAL USING POINTERS.

78) WRITE A PROGRAM TO IMPLEMENT A CIRCULAR QUEUE USING POINTERS.

79) WRITE A PROGRAM TO MERGE TWO SORTED LINKED LISTS INTO A SINGLE SORTED LIST USING POINTERS.

80) WRITE A PROGRAM TO ROTATE A LINKED LIST TO THE RIGHT BY K PLACES USING POINTERS.

81) WRITE A PROGRAM TO DELETE EVERY N-TH NODE IN A LINKED LIST USING POINTERS.

82) WRITE A PROGRAM TO FIND THE INTERSECTION OF TWO ARRAYS USING POINTERS.

83) WRITE A PROGRAM TO IMPLEMENT A SINGLY LINKED LIST WITH NODES INSERTED IN SORTED ORDER USING POINTERS.

84) WRITE A PROGRAM TO CHECK FOR A LOOP IN A LINKED LIST USING POINTERS.

85) WRITE A PROGRAM TO COPY A MULTIDIMENSIONAL ARRAY USING POINTERS.

86) WRITE A PROGRAM TO IMPLEMENT MEMORY POOLING USING POINTERS.

87) WRITE A PROGRAM TO FIND THE LONGEST PALINDROMIC SUBSTRING IN A STRING USING POINTERS.

88) WRITE A PROGRAM TO DELETE ALL OCCURRENCES OF A GIVEN ELEMENT FROM A LINKED LIST USING POINTERS.

89) WRITE A PROGRAM TO CONVERT A BINARY TREE INTO A DOUBLY LINKED LIST USING POINTERS.

90) WRITE A PROGRAM TO IMPLEMENT DEPTH-FIRST SEARCH (DFS) IN A GRAPH USING POINTERS.

91) WRITE A PROGRAM TO IMPLEMENT BREADTH-FIRST SEARCH (BFS) IN A GRAPH USING POINTERS.

92) WRITE A PROGRAM TO FIND THE DIFFERENCE BETWEEN TWO LINKED LISTS USING POINTERS.

93) WRITE A PROGRAM TO CHECK IF TWO STRINGS ARE ANAGRAMS USING POINTERS.

94) WRITE A PROGRAM TO IMPLEMENT MATRIX MULTIPLICATION USING DYNAMIC MEMORY ALLOCATION AND POINTERS.

95) WRITE A PROGRAM TO FIND THE K-TH SMALLEST ELEMENT IN AN ARRAY USING POINTERS.

96) WRITE A PROGRAM TO CHECK IF A BINARY TREE IS A FULL BINARY TREE USING POINTERS.

97) WRITE A PROGRAM TO CREATE A CIRCULAR DOUBLY LINKED LIST USING POINTERS.

98) WRITE A PROGRAM TO SORT AN ARRAY USING QUICK SORT WITH POINTERS.

99) WRITE A PROGRAM TO IMPLEMENT A HEAP DATA STRUCTURE USING POINTERS.

100) WRITE A PROGRAM TO CREATE A MIN-HEAP AND MAX-HEAP FROM AN ARRAY USING POINTERS.

101) WRITE A PROGRAM TO FIND THE LONGEST CONSECUTIVE SEQUENCE IN AN ARRAY USING POINTERS.

102) WRITE A PROGRAM TO PERFORM INSERTION SORT ON A LINKED LIST USING POINTERS.

103) WRITE A PROGRAM TO DETECT AND REMOVE A LOOP IN A LINKED LIST USING POINTERS.

104) WRITE A PROGRAM TO FIND THE NEXT LARGER NODE FOR EACH NODE IN A LINKED LIST USING POINTERS.

105) WRITE A PROGRAM TO IMPLEMENT CIRCULAR BUFFER USING POINTERS.

106) WRITE A PROGRAM TO CREATE AND DISPLAY A TRIANGULAR MATRIX USING POINTERS.

107) WRITE A PROGRAM TO IMPLEMENT A TRIE DATA STRUCTURE USING POINTERS.

108) WRITE A PROGRAM TO PERFORM MERGE OPERATION ON TWO BALANCED BINARY SEARCH TREES USING POINTERS.

109) WRITE A PROGRAM TO FIND THE LEVEL ORDER TRAVERSAL OF A BINARY TREE USING POINTERS.

110) WRITE A PROGRAM TO IMPLEMENT DIJKSTRA'S ALGORITHM FOR SHORTEST PATH USING POINTERS.

111) WRITE A PROGRAM TO IMPLEMENT FLOYD-WARSHALL ALGORITHM FOR ALL PAIRS SHORTEST PATH USING POINTERS.

112) WRITE A PROGRAM TO REVERSE EACH WORD IN A STRING USING POINTERS.

113) WRITE A PROGRAM TO FIND THE KTH LARGEST ELEMENT IN A LINKED LIST USING POINTERS.

114) WRITE A PROGRAM TO PERFORM MEMORY COMPACTION USING POINTERS.

115) WRITE A PROGRAM TO FIND THE MAXIMUM SUM PATH BETWEEN TWO LEAVES IN A BINARY TREE USING POINTERS.

116) WRITE A PROGRAM TO CONVERT A SINGLY LINKED LIST TO A CIRCULAR LINKED LIST USING POINTERS.

117) WRITE A PROGRAM TO IMPLEMENT MATRIX ROTATION (90 DEGREES CLOCKWISE) USING POINTERS.

118) WRITE A PROGRAM TO FIND THE MAXIMUM DEPTH OF A N-ARY TREE USING POINTERS.

119) WRITE A PROGRAM TO DELETE DUPLICATE CHARACTERS IN A STRING USING POINTERS.

120) WRITE A PROGRAM TO COUNT FREQUENCY OF CHARACTERS IN A STRING USING POINTERS.

=> 17. MATRIX

1) WRITE A PROGRAM TO TRANSPOSE A 2D ARRAY OR MATRIX.

2) WRITE A PROGRAM TO FIND THE ROW WITH THE MAXIMUM NUMBER OF 1S IN A BINARY MATRIX.

3) WRITE A PROGRAM TO ROTATE A 2D ARRAY 90 DEGREES CLOCKWISE.

4) WRITE A PROGRAM TO CALCULATE THE BOUNDARY ELEMENT SUM OF A 2D MATRIX.

5) WRITE A PROGRAM TO FIND THE LONGEST PATH OF CONSECUTIVE ELEMENTS IN A 2D ARRAY.

6) WRITE A PROGRAM TO CHECK IF A 2D ARRAY IS SYMMETRIC.

7) WRITE A PROGRAM TO PRINT A 2D ARRAY IN A SPIRAL ORDER.

8) WRITE A PROGRAM TO FIND THE SADDLE POINT (MIN ROW, MAX COLUMN) IN A MATRIX.

9) WRITE A PROGRAM TO FIND ALL POSSIBLE RECTANGLES FORMED WITHIN A 2D ARRAY.

10) WRITE A PROGRAM TO FIND THE TOTAL NUMBER OF SQUARE SUB-MATRICES IN A MATRIX.

11) WRITE A PROGRAM TO CHECK IF A 2D ARRAY REPRESENTS A MAGIC SQUARE.

12) WRITE A PROGRAM TO ROTATE A 2D ARRAY COUNTERCLOCKWISE BY 180 DEGREES.

13) WRITE A PROGRAM TO FIND THE ROW AND COLUMN INDEXES OF ALL PEAK ELEMENTS IN A MATRIX.

14) WRITE A PROGRAM TO ADD TWO MATRICES ELEMENT-WISE.

15) WRITE A PROGRAM TO FIND THE TRACE (SUM OF MAIN DIAGONAL ELEMENTS) OF A MATRIX.

16) WRITE A PROGRAM TO MULTIPLY TWO MATRICES.

17) WRITE A PROGRAM TO CONVERT A 2D MATRIX TO ITS LOWER TRIANGULAR FORM.

18) WRITE A PROGRAM TO CONVERT A 2D MATRIX TO ITS UPPER TRIANGULAR FORM.

19) WRITE A PROGRAM TO FIND THE MAXIMUM SUM OF AN HOURGLASS IN A MATRIX.

20) WRITE A PROGRAM TO FIND ALL ELEMENTS IN A 2D ARRAY THAT ARE GREATER THAN A GIVEN THRESHOLD.

21) WRITE A PROGRAM TO TRANSPOSE ONLY THE UPPER TRIANGLE OF A MATRIX.

22) WRITE A PROGRAM TO CHECK IF ALL DIAGONALS OF A MATRIX HAVE THE SAME ELEMENT.

23) WRITE A PROGRAM TO FIND THE SMALLEST ELEMENT IN EACH ROW OF A MATRIX.

24) WRITE A PROGRAM TO CREATE A SNAKE-LIKE PATTERN OF A MATRIX.

25) WRITE A PROGRAM TO IMPLEMENT MATRIX CHAIN MULTIPLICATION.

=> 18. RECURSION

1) WRITE A PROGRAM TO FIND THE FACTORIAL OF A NUMBER USING RECURSION.

2) WRITE A PROGRAM TO COMPUTE THE FIBONACCI SERIES UP TO N TERMS USING RECURSION.

3) WRITE A PROGRAM TO FIND THE SUM OF N NATURAL NUMBERS USING RECURSION.

4) WRITE A PROGRAM TO CALCULATE THE POWER OF A NUMBER USING RECURSION.

5) WRITE A PROGRAM TO FIND THE GREATEST COMMON DIVISOR (GCD) OF TWO NUMBERS USING RECURSION.

6) WRITE A PROGRAM TO COMPUTE THE LEAST COMMON MULTIPLE (LCM) OF TWO NUMBERS USING RECURSION.

7) WRITE A PROGRAM TO REVERSE A STRING USING RECURSION.

8) WRITE A PROGRAM TO CALCULATE THE SUM OF DIGITS OF A NUMBER USING RECURSION.

9) WRITE A PROGRAM TO CHECK IF A NUMBER IS A PALINDROME USING RECURSION.

10) WRITE A PROGRAM TO COUNT THE NUMBER OF DIGITS IN A NUMBER USING RECURSION.

11) WRITE A PROGRAM TO FIND THE SUM OF ELEMENTS IN AN ARRAY USING RECURSION.

12) WRITE A PROGRAM TO SEARCH FOR AN ELEMENT IN AN ARRAY USING RECURSION.

13) WRITE A PROGRAM TO COMPUTE THE PRODUCT OF TWO NUMBERS USING RECURSION.

14) WRITE A PROGRAM TO CHECK IF A GIVEN STRING IS A PALINDROME USING RECURSION.

15) WRITE A PROGRAM TO FIND THE NTH TRIANGULAR NUMBER USING RECURSION.

16) WRITE A PROGRAM TO CALCULATE THE SUM OF THE FIRST N EVEN NUMBERS USING RECURSION.

17) WRITE A PROGRAM TO CALCULATE THE SUM OF THE FIRST N ODD NUMBERS USING RECURSION.

18) WRITE A PROGRAM TO FIND THE MINIMUM ELEMENT IN AN ARRAY USING RECURSION.

19) WRITE A PROGRAM TO FIND THE MAXIMUM ELEMENT IN AN ARRAY USING RECURSION.

20) WRITE A PROGRAM TO SORT AN ARRAY USING RECURSION (MERGE SORT).

21) WRITE A PROGRAM TO IMPLEMENT THE BINARY SEARCH ALGORITHM USING RECURSION.

22) WRITE A PROGRAM TO CONVERT A DECIMAL NUMBER TO BINARY USING RECURSION.

23) WRITE A PROGRAM TO CONVERT A DECIMAL NUMBER TO OCTAL USING RECURSION.

24) WRITE A PROGRAM TO CONVERT A DECIMAL NUMBER TO HEXADECIMAL USING RECURSION.

25) WRITE A PROGRAM TO CALCULATE THE NTH HARMONIC NUMBER USING RECURSION.

26) WRITE A PROGRAM TO FIND THE PERMUTATIONS OF A STRING USING RECURSION.

27) WRITE A PROGRAM TO FIND THE COMBINATIONS OF ELEMENTS IN AN ARRAY USING RECURSION.

28) WRITE A PROGRAM TO SOLVE THE TOWER OF HANOI PROBLEM USING RECURSION.

29) WRITE A PROGRAM TO FIND THE SUM OF A GEOMETRIC SERIES USING RECURSION.

30) WRITE A PROGRAM TO COMPUTE THE EXPONENTIAL FUNCTION USING RECURSION.

31) WRITE A PROGRAM TO GENERATE ALL SUBSETS OF A SET USING RECURSION.

32) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE VERSION OF THE SELECTION SORT ALGORITHM.

33) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE VERSION OF THE BUBBLE SORT ALGORITHM.

34) WRITE A PROGRAM TO REVERSE AN ARRAY USING RECURSION.

35) WRITE A PROGRAM TO FIND THE LENGTH OF A STRING USING RECURSION.

36) WRITE A PROGRAM TO COUNT THE OCCURRENCES OF A CHARACTER IN A STRING USING RECURSION.

37) WRITE A PROGRAM TO FIND THE SUM OF ALL ELEMENTS IN A MATRIX USING RECURSION.

38) WRITE A PROGRAM TO COMPUTE THE PRODUCT OF ALL ELEMENTS IN AN ARRAY USING RECURSION.

39) WRITE A PROGRAM TO FIND THE NTH UGLY NUMBER USING RECURSION.

40) WRITE A PROGRAM TO CALCULATE BINOMIAL COEFFICIENTS USING RECURSION.

41) WRITE A PROGRAM TO FIND THE MAXIMUM DEPTH OF A BINARY TREE USING RECURSION.

42) WRITE A PROGRAM TO COUNT THE NUMBER OF LEAF NODES IN A BINARY TREE USING RECURSION.

43) WRITE A PROGRAM TO CHECK IF TWO STRINGS ARE ANAGRAMS USING RECURSION.

44) WRITE A PROGRAM TO COUNT THE NUMBER OF VOWELS IN A STRING USING RECURSION.

45) WRITE A PROGRAM TO COUNT THE NUMBER OF CONSONANTS IN A STRING USING RECURSION.

46) WRITE A PROGRAM TO PRINT THE REVERSE OF A LINKED LIST USING RECURSION.

47) WRITE A PROGRAM TO COMPUTE THE NTH CATALAN NUMBER USING RECURSION.

48) WRITE A PROGRAM TO FIND THE NTH LUCAS NUMBER USING RECURSION.

49) WRITE A PROGRAM TO COMPUTE THE SUM OF A SERIES USING RECURSION ($1/1! + 1/2! + \dots$).

50) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE DEPTH-FIRST SEARCH (DFS) ALGORITHM FOR GRAPHS.

51) WRITE A PROGRAM TO FIND ALL POSSIBLE DECODINGS OF A NUMBER STRING USING RECURSION (LIKE IN ENCODING $A=1$, $B=2$).

52) WRITE A PROGRAM TO PRINT ALL POSSIBLE PARENTHESES COMBINATIONS FOR N PAIRS USING RECURSION.

53) WRITE A PROGRAM TO CALCULATE THE DIGITAL ROOT OF A NUMBER USING RECURSION.

54) WRITE A PROGRAM TO FIND ALL PALINDROMIC PARTITIONS OF A STRING USING RECURSION.

55) WRITE A PROGRAM TO FIND ALL POSSIBLE ARRANGEMENTS OF A NUMBER USING RECURSION.

56) WRITE A PROGRAM TO CHECK IF A NUMBER IS A PERFECT NUMBER USING RECURSION.

57) WRITE A PROGRAM TO GENERATE ALL COMBINATIONS OF A STRING USING RECURSION.

58) WRITE A PROGRAM TO FIND THE NUMBER OF WAYS TO CLIMB N STAIRS WITH 1 OR 2 STEPS USING RECURSION.

59) WRITE A PROGRAM TO SOLVE THE N-QUEENS PROBLEM USING RECURSION.

60) WRITE A PROGRAM TO SOLVE THE KNAPSACK PROBLEM USING RECURSION.

61) WRITE A PROGRAM TO FIND THE NTH NUMBER IN A PADOVAN SEQUENCE USING RECURSION.

62) WRITE A PROGRAM TO SOLVE THE SUDOKU PUZZLE USING RECURSION.

63) WRITE A PROGRAM TO FIND THE LONGEST COMMON SUBSEQUENCE (LCS) OF TWO STRINGS USING RECURSION.

64) WRITE A PROGRAM TO FIND ALL POSSIBLE SUMS OF SUBSETS OF AN ARRAY USING RECURSION.

65) WRITE A PROGRAM TO PRINT ALL PERMUTATIONS OF AN ARRAY USING RECURSION.

66) WRITE A PROGRAM TO CALCULATE THE NTH HEXAGONAL NUMBER USING RECURSION.

67) WRITE A PROGRAM TO PRINT ALL PATHS FROM TOP-LEFT TO BOTTOM-RIGHT OF A GRID USING RECURSION.

68) WRITE A PROGRAM TO COMPUTE THE NTH TETRAHEDRAL NUMBER USING RECURSION.

69) WRITE A PROGRAM TO IMPLEMENT THE KNUTH-MORRIS-PRATT (KMP) ALGORITHM USING RECURSION.

70) WRITE A PROGRAM TO CALCULATE THE NTH HEXADECIMAL NUMBER USING RECURSION.

71) WRITE A PROGRAM TO COMPUTE ALL UNIQUE PERMUTATIONS OF AN ARRAY WITH DUPLICATES USING RECURSION.

72) WRITE A PROGRAM TO GENERATE THE NTH HAPPY NUMBER USING RECURSION.

73) WRITE A PROGRAM TO SOLVE THE JUGGLER SEQUENCE PROBLEM USING RECURSION.

74) WRITE A PROGRAM TO COMPUTE THE DECIMAL TO ROMAN CONVERSION USING RECURSION.

75) WRITE A PROGRAM TO SOLVE THE LIGHT BULB SWITCHER PROBLEM USING RECURSION.

76) WRITE A PROGRAM TO CALCULATE THE NTH STELLA OCTANGULA NUMBER USING RECURSION.

77) WRITE A PROGRAM TO FIND ALL PALINDROMIC SUBSEQUENCES IN A STRING USING RECURSION.

78) WRITE A PROGRAM TO PRINT ALL UNIQUE COMBINATIONS SUMMING TO A TARGET VALUE USING RECURSION.

79) WRITE A PROGRAM TO FIND THE TOTAL NUMBER OF PATHS TO REACH A TARGET NUMBER BY ADDING 1, 2, OR 3 USING RECURSION.

80) WRITE A PROGRAM TO COMPUTE THE COUNT OF POSSIBLE BINARY SEARCH TREES WITH N NODES USING RECURSION.

81) WRITE A PROGRAM TO PRINT ALL PATHS FROM ROOT TO LEAF NODES IN A BINARY TREE USING RECURSION.

82) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE FUNCTION FOR TOWER OF HANOI WITH FOUR POLES.

83) WRITE A PROGRAM TO PRINT ALL POSSIBLE BRACKET COMBINATIONS FOR N PAIRS USING RECURSION.

84) WRITE A PROGRAM TO COMPUTE THE NTH PENTAGONAL NUMBER USING RECURSION.

85) WRITE A PROGRAM TO COUNT THE NUMBER OF WAYS TO PARTITION A NUMBER USING RECURSION.

86) WRITE A PROGRAM TO IMPLEMENT RECURSIVE BINARY TREE LEVEL ORDER PRINTING.

87) WRITE A PROGRAM TO CALCULATE THE SUM OF DIGITS UNTIL A SINGLE DIGIT IS LEFT USING RECURSION.

88) WRITE A PROGRAM TO FIND ALL COMBINATIONS THAT SUM TO A GIVEN TARGET VALUE USING RECURSION.

89) WRITE A PROGRAM TO COMPUTE THE EXPRESSION RESULT OF N TERMS IN AN ARITHMETIC SERIES USING RECURSION.

90) WRITE A PROGRAM TO FIND ALL PATHS TO CLIMB STAIRS WITH 1, 2, OR 3 STEPS USING RECURSION.

91) WRITE A PROGRAM TO SOLVE THE CHESS KNIGHT TOUR PROBLEM USING RECURSION.

92) WRITE A PROGRAM TO GENERATE ALL PALINDROMIC STRINGS FROM A GIVEN STRING USING RECURSION.

93) WRITE A PROGRAM TO FIND ALL UNIQUE BINARY TREES THAT CAN BE FORMED WITH N NODES USING RECURSION.

94) WRITE A PROGRAM TO GENERATE ALL POSSIBLE ARRANGEMENTS OF A SET OF NUMBERS USING RECURSION.

95) WRITE A PROGRAM TO CALCULATE THE TOTAL NUMBER OF DISTINCT PARTITIONS OF A GIVEN NUMBER USING RECURSION.

96) WRITE A PROGRAM TO FIND THE LONGEST PALINDROMIC SUBSEQUENCE IN A STRING USING RECURSION.

97) WRITE A PROGRAM TO CALCULATE THE NTH SUPER CATALAN NUMBER USING RECURSION.

98) WRITE A PROGRAM TO FIND THE COUNT OF SUBSEQUENCES THAT SUM TO A SPECIFIC VALUE USING RECURSION.

99) WRITE A PROGRAM TO COMPUTE THE PARTITION FUNCTION FOR INTEGERS USING RECURSION.

100) WRITE A PROGRAM TO COUNT ALL DISTINCT BINARY TREES WITH N DISTINCT NODES USING RECURSION.

101) WRITE A PROGRAM TO CALCULATE THE NTH ZIGZAG NUMBER USING RECURSION.

102) WRITE A PROGRAM TO PRINT ALL POSSIBLE BINARY STRINGS OF A GIVEN LENGTH USING RECURSION.

103) WRITE A PROGRAM TO FIND THE MAXIMUM PRODUCT OF BREAKING A NUMBER INTO INTEGER SUMS USING RECURSION.

104) WRITE A PROGRAM TO CALCULATE THE NTH RECURRING TERNARY NUMBER USING RECURSION.

105) WRITE A PROGRAM TO PRINT ALL POSSIBLE PALINDROMES IN A STRING USING RECURSION.

106) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE SOLVER FOR THE SUDOKU PUZZLE WITH BACKTRACKING.

107) WRITE A PROGRAM TO GENERATE ALL ARRANGEMENTS OF BRACKETS WITH N PAIRS USING RECURSION.

108) WRITE A PROGRAM TO CALCULATE ALL POSSIBLE SUM COMBINATIONS OF A TARGET VALUE USING RECURSION.

109) WRITE A PROGRAM TO COUNT THE NUMBER OF DISTINCT ARRANGEMENTS OF LETTERS IN A STRING USING RECURSION.

110) WRITE A PROGRAM TO FIND THE MAXIMUM LENGTH OF A CHAIN OF STRING LINKS USING RECURSION.

111) WRITE A PROGRAM TO SOLVE THE RAT IN A MAZE PROBLEM USING RECURSION.

112) WRITE A PROGRAM TO PRINT ALL DISTINCT ARRANGEMENTS OF CHARACTERS IN A STRING USING RECURSION.

113) WRITE A PROGRAM TO FIND ALL COMBINATIONS OF A BINARY STRING THAT BALANCE BRACKETS USING RECURSION.

114) WRITE A PROGRAM TO COUNT ALL WAYS TO FILL A $N \times M$ GRID USING TILES OF FIXED SIZE USING RECURSION.

115) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE SOLVER FOR MINIMUM COIN CHANGE PROBLEM.

116) WRITE A PROGRAM TO FIND THE LONGEST PATH IN A MATRIX USING RECURSION.

117) WRITE A PROGRAM TO GENERATE A MULTIPLICATION TABLE USING RECURSION.

118) WRITE A PROGRAM TO FIND ALL UNIQUE N-QUEENS SOLUTIONS ON AN $N \times N$ BOARD USING RECURSION.

119) WRITE A PROGRAM TO COMPUTE THE SUM OF NUMBERS DIVISIBLE BY A SPECIFIC DIVISOR IN A RANGE USING RECURSION.

120) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE FUNCTION FOR DEPTH-FIRST SEARCH IN A GRAPH.

121) WRITE A PROGRAM TO COMPUTE THE TETRAHEDRAL NUMBERS FOR A GIVEN N USING RECURSION.

122) WRITE A PROGRAM TO CALCULATE THE NTH STELLA OCTANGULA NUMBER USING RECURSION.

123) WRITE A PROGRAM TO FIND ALL PALINDROMIC SUBSEQUENCES IN A STRING USING RECURSION.

124) WRITE A PROGRAM TO PRINT ALL UNIQUE COMBINATIONS SUMMING TO A TARGET VALUE USING RECURSION.

125) WRITE A PROGRAM TO FIND THE TOTAL NUMBER OF PATHS TO REACH A TARGET NUMBER BY ADDING 1, 2, OR 3 USING RECURSION.

126) WRITE A PROGRAM TO COMPUTE THE COUNT OF POSSIBLE BINARY SEARCH TREES WITH N NODES USING RECURSION.

127) WRITE A PROGRAM TO PRINT ALL PATHS FROM ROOT TO LEAF NODES IN A BINARY TREE USING RECURSION.

128) WRITE A PROGRAM TO IMPLEMENT A RECURSIVE FUNCTION FOR TOWER OF HANOI WITH FOUR POLES.

129) WRITE A PROGRAM TO PRINT ALL POSSIBLE BRACKET COMBINATIONS FOR N PAIRS USING RECURSION.

130) WRITE A PROGRAM TO COMPUTE THE NTH PENTAGONAL NUMBER USING RECURSION.

131) WRITE A PROGRAM TO COUNT THE NUMBER OF WAYS TO PARTITION A NUMBER USING RECURSION.

132) WRITE A PROGRAM TO IMPLEMENT RECURSIVE BINARY TREE LEVEL ORDER PRINTING.

133) WRITE A PROGRAM TO CALCULATE THE SUM OF DIGITS UNTIL A SINGLE DIGIT IS LEFT USING RECURSION.

134) WRITE A PROGRAM TO FIND ALL COMBINATIONS THAT SUM TO A GIVEN TARGET VALUE USING RECURSION.

135) WRITE A PROGRAM TO COMPUTE THE EXPRESSION RESULT OF N TERMS IN AN ARITHMETIC SERIES USING RECURSION.

136) WRITE A PROGRAM TO FIND ALL PATHS TO CLIMB STAIRS WITH 1, 2, OR 3 STEPS USING RECURSION.

137) WRITE A PROGRAM TO SOLVE THE CHESS KNIGHT TOUR PROBLEM USING RECURSION.

138) WRITE A PROGRAM TO GENERATE ALL PALINDROMIC STRINGS FROM A GIVEN STRING USING RECURSION.

139) WRITE A PROGRAM TO FIND ALL UNIQUE BINARY TREES THAT CAN BE FORMED WITH N NODES USING RECURSION.

140) WRITE A PROGRAM TO GENERATE ALL POSSIBLE ARRANGEMENTS OF A SET OF NUMBERS USING RECURSION.

141) WRITE A PROGRAM TO CALCULATE THE TOTAL NUMBER OF DISTINCT PARTITIONS OF A GIVEN NUMBER USING RECURSION.

142) WRITE A PROGRAM TO FIND THE LONGEST PALINDROMIC SUBSEQUENCE IN A STRING USING RECURSION.

143) WRITE A PROGRAM TO CALCULATE THE NTH SUPER CATALAN NUMBER USING RECURSION.

144) WRITE A PROGRAM TO FIND THE COUNT OF SUBSEQUENCES THAT SUM TO A SPECIFIC VALUE USING RECURSION.

145) WRITE A PROGRAM TO COUNT THE NUMBER OF WAYS TO REACH THE NTH STAIR WITH RESTRICTIONS USING RECURSION.

146) WRITE A PROGRAM TO SOLVE THE SET PARTITION PROBLEM BY DIVIDING ELEMENTS INTO TWO SUBSETS OF EQUAL SUM USING RECURSION.

147) WRITE A PROGRAM TO FIND ALL POSSIBLE DECODINGS OF A DIGIT STRING REPRESENTING ENGLISH LETTERS (1-26) USING RECURSION.

148) WRITE A PROGRAM TO COUNT ALL POSSIBLE UNIQUE PATHS IN A TRIANGULAR GRID USING RECURSION.

149) WRITE A PROGRAM TO GENERATE ALL UNIQUE BINARY STRINGS OF LENGTH N WITH NO TWO CONSECUTIVE ONES USING RECURSION.

150) WRITE A PROGRAM TO FIND ALL POSSIBLE PALINDROMIC PARTITIONS OF A STRING USING RECURSION.

151) WRITE A PROGRAM TO SOLVE THE COUNT OF SUBSETS WITH A GIVEN DIFFERENCE USING RECURSION.

152) WRITE A PROGRAM TO FIND ALL POSSIBLE NUMBERS BY REARRANGING DIGITS OF A NUMBER USING RECURSION.

153) WRITE A PROGRAM TO GENERATE ALL VALID IP ADDRESSES FROM A STRING OF DIGITS USING RECURSION.

154) WRITE A PROGRAM TO SOLVE THE N-ROOKS PROBLEM ON AN $N \times N$ CHESSBOARD USING RECURSION.

155) WRITE A PROGRAM TO COUNT ALL PATHS IN A MAZE WITH OBSTACLES USING RECURSION.

156) WRITE A PROGRAM TO GENERATE ALL POSSIBLE ARRANGEMENTS OF A 3-COLOR FLAG USING RECURSION.

157) WRITE A PROGRAM TO FIND ALL POSSIBLE PARTITIONS OF A STRING INTO PALINDROMIC SUBSTRINGS USING RECURSION.

158) WRITE A PROGRAM TO COUNT THE NUMBER OF UNIQUE WAYS TO SPLIT AN ARRAY OF N ELEMENTS INTO TWO SUBSETS WITH EQUAL SUM USING RECURSION.

159) WRITE A PROGRAM TO GENERATE ALL COMBINATIONS OF K ELEMENTS FROM A SET OF SIZE N USING RECURSION.

160) WRITE A PROGRAM TO FIND ALL UNIQUE FACTORIZATIONS OF A NUMBER USING RECURSION.

161) WRITE A PROGRAM TO SOLVE THE MINIMUM CUT PALINDROME PARTITIONING OF A STRING USING RECURSION.

162) WRITE A PROGRAM TO GENERATE ALL POSSIBLE WAYS TO PLACE K KINGS ON A CHESSBOARD USING RECURSION.

163) WRITE A PROGRAM TO FIND ALL UNIQUE SUBSETS OF A GIVEN SET USING RECURSION.

164) WRITE A PROGRAM TO SOLVE THE 0-1 KNAPSACK PROBLEM USING RECURSION.

165) WRITE A PROGRAM TO COUNT ALL POSSIBLE PATHS FROM THE TOP-LEFT TO BOTTOM-RIGHT CORNER OF A GRID USING RECURSION.

166) WRITE A PROGRAM TO SOLVE THE SUM OF K NUMBERS PROBLEM THAT EQUALS TO N USING RECURSION.

167) WRITE A PROGRAM TO GENERATE ALL PERMUTATIONS OF A STRING WITH CASE CHANGES USING RECURSION.

168) WRITE A PROGRAM TO FIND THE MAXIMUM NUMBER OF NON-OVERLAPPING PALINDROMIC SUBSTRINGS IN A STRING USING RECURSION.

169) WRITE A PROGRAM TO COUNT ALL VALID NUMERIC EXPRESSIONS THAT EVALUATE TO A TARGET VALUE USING RECURSION.

170) WRITE A PROGRAM TO GENERATE ALL WAYS TO COVER A $2 \times N$ BOARD USING 1×2 AND 2×1 TILES USING RECURSION.

171) WRITE A PROGRAM TO SOLVE THE SCRAMBLE STRING PROBLEM USING RECURSION.

172) WRITE A PROGRAM TO FIND THE NUMBER OF WAYS TO ASSIGN PLUS AND MINUS SIGNS TO MAKE THE SUM OF ELEMENTS EQUAL A TARGET VALUE USING RECURSION.

173) WRITE A PROGRAM TO FIND ALL POSSIBLE COMBINATIONS OF NUMBERS THAT ADD UP TO A TARGET VALUE USING RECURSION.

174) WRITE A PROGRAM TO GENERATE ALL POSSIBLE WAYS TO FILL A $N \times M$ GRID USING DOMINOS OF SIZE 1×2 USING RECURSION.

175) WRITE A PROGRAM TO CALCULATE THE NUMBER OF WAYS TO FORM A GIVEN SUM USING NUMBERS 1 TO K USING RECURSION.

176) WRITE A PROGRAM TO FIND ALL POSSIBLE K-PARTITIONS OF A STRING USING RECURSION.

177) WRITE A PROGRAM TO SOLVE THE MAXIMUM CUTTING OF A ROD INTO SEGMENTS OF GIVEN LENGTHS USING RECURSION.

178) WRITE A PROGRAM TO GENERATE ALL POSSIBLE ARRANGEMENTS OF A SET OF DISTINCT ITEMS WHERE ORDER MATTERS USING RECURSION.

179) WRITE A PROGRAM TO FIND ALL UNIQUE COMBINATIONS OF CHARACTERS FROM A STRING USING RECURSION.

180) WRITE A PROGRAM TO SOLVE THE MINIMUM COST PATH PROBLEM IN A GRID USING RECURSION.

181) WRITE A PROGRAM TO GENERATE ALL POSSIBLE BINARY SEQUENCES WITH EXACTLY K BITS SET USING RECURSION.

182) WRITE A PROGRAM TO COUNT ALL WAYS TO SPLIT AN ARRAY INTO THREE PARTS WITH EQUAL SUM USING RECURSION.

183) WRITE A PROGRAM TO FIND ALL COMBINATIONS OF COINS THAT MAKE UP A GIVEN AMOUNT USING RECURSION.

184) WRITE A PROGRAM TO GENERATE ALL UNIQUE PATHS IN A 3D MAZE USING RECURSION.

185) WRITE A PROGRAM TO FIND THE MINIMUM NUMBER OF COINS NEEDED TO MAKE UP A TARGET VALUE USING RECURSION.

186) WRITE A PROGRAM TO COUNT THE NUMBER OF WAYS TO TILE A $2 \times N$ FLOOR USING 2×1 AND 1×2 TILES USING RECURSION.

187) WRITE A PROGRAM TO GENERATE ALL VALID ARRANGEMENTS OF PARENTHESES FOR N PAIRS USING RECURSION.

188) WRITE A PROGRAM TO SOLVE THE STRING INTERLEAVING PROBLEM USING RECURSION.

189) WRITE A PROGRAM TO FIND ALL POSSIBLE SEQUENCES OF SUMS THAT RESULT IN A GIVEN NUMBER USING RECURSION.

190) WRITE A PROGRAM TO GENERATE ALL UNIQUE PERMUTATIONS OF AN ARRAY WITH DUPLICATE ELEMENTS USING RECURSION.

191) WRITE A PROGRAM TO SOLVE THE K-SUM PARTITION PROBLEM WHERE THE ARRAY IS SPLIT INTO K SUBSETS OF EQUAL SUM USING RECURSION.

192) WRITE A PROGRAM TO FIND THE NUMBER OF WAYS TO REACH A GIVEN SCORE IN A GAME USING RECURSION.

193) WRITE A PROGRAM TO GENERATE ALL SUBSEQUENCES OF A STRING THAT HAVE A GIVEN SUM OF ASCII VALUES USING RECURSION.

194) WRITE A PROGRAM TO COUNT ALL UNIQUE PARTITIONS OF AN INTEGER USING RECURSION.

195) WRITE A PROGRAM TO FIND ALL POSSIBLE BINARY SEARCH TREES WITH N NODES USING RECURSION.

196) WRITE A PROGRAM TO FIND ALL COMBINATIONS OF ARRAY ELEMENTS THAT HAVE A PRODUCT EQUAL TO A TARGET VALUE USING RECURSION.

197) WRITE A PROGRAM TO SOLVE THE SCRAMBLE STRING PROBLEM USING MEMOIZATION AND RECURSION.

198) WRITE A PROGRAM TO COUNT ALL POSSIBLE PALINDROMIC SUBSTRINGS IN A GIVEN STRING USING RECURSION.

199) WRITE A PROGRAM TO FIND THE LONGEST PATH FROM THE TOP LEFT TO THE BOTTOM RIGHT CORNER IN A GRID USING RECURSION.

200) WRITE A PROGRAM TO GENERATE ALL POSSIBLE COMBINATIONS OF N STRINGS USING RECURSION.

201) WRITE A PROGRAM TO SOLVE THE SUM OF ALL PATHS FROM ROOT TO LEAVES IN A BINARY TREE USING RECURSION.

202) WRITE A PROGRAM TO FIND ALL K-COMBINATION SUBSETS OF A SET USING RECURSION.

203) WRITE A PROGRAM TO COUNT ALL WAYS TO SPLIT AN ARRAY INTO N PARTS WITH EQUAL SUM USING RECURSION.

204) WRITE A PROGRAM TO GENERATE ALL UNIQUE COMBINATIONS OF CHARACTERS IN A STRING WHERE DUPLICATES ARE NOT ALLOWED USING RECURSION.

205) WRITE A PROGRAM TO SOLVE THE MINIMUM NUMBER OF CUTS NEEDED TO PARTITION A STRING INTO PALINDROMES USING RECURSION.

206) WRITE A PROGRAM TO FIND ALL PERMUTATIONS OF A LIST OF NUMBERS THAT MEET A GIVEN CONDITION USING RECURSION.

207) WRITE A PROGRAM TO GENERATE ALL POSSIBLE VALID BRACKETING FOR A GIVEN NUMBER OF BRACKETS USING RECURSION.

208) WRITE A PROGRAM TO FIND ALL PATHS IN A MAZE WHERE MOVES ARE LIMITED TO ONLY DOWN OR RIGHT USING RECURSION.

209) WRITE A PROGRAM TO COUNT THE NUMBER OF DISTINCT WAYS TO ASSIGN BINARY VALUES TO ELEMENTS IN AN ARRAY TO MEET A TARGET SUM USING RECURSION.

210) WRITE A PROGRAM TO GENERATE ALL POSSIBLE COMBINATIONS OF ITEMS IN A MULTISSET USING RECURSION.

211) WRITE A PROGRAM TO SOLVE THE MAXIMUM SCORE FROM COLLECTING COINS IN A MAZE USING RECURSION.

212) WRITE A PROGRAM TO FIND ALL POSSIBLE ARRANGEMENTS OF K DISTINCT ITEMS OUT OF N USING RECURSION.

213) WRITE A PROGRAM TO COUNT ALL UNIQUE PATHS IN AN OBSTACLE-FILLED GRID USING RECURSION.

214) WRITE A PROGRAM TO FIND ALL UNIQUE WAYS TO PLACE N KNIGHTS ON AN N X N CHESSBOARD USING RECURSION.

215) WRITE A PROGRAM TO GENERATE ALL SUBSETS OF AN ARRAY THAT HAVE A PRODUCT LESS THAN A GIVEN VALUE USING RECURSION.

216) WRITE A PROGRAM TO SOLVE THE TARGET SUM SUBSET PROBLEM USING RECURSION.

217) WRITE A PROGRAM TO COUNT ALL POSSIBLE WAYS TO ARRANGE PARENTHESES TO EVALUATE TO A GIVEN BOOLEAN RESULT USING RECURSION.

218) WRITE A PROGRAM TO GENERATE ALL PERMUTATIONS OF A STRING THAT START WITH A VOWEL USING RECURSION.

219) WRITE A PROGRAM TO FIND THE LONGEST COMMON SUBSEQUENCE BETWEEN TWO STRINGS USING RECURSION.

220) WRITE A PROGRAM TO SOLVE THE MINIMUM PATH SUM PROBLEM IN A TRIANGULAR ARRAY USING RECURSION.

=> 19. MINI PROJECTS

1) WRITE A PROGRAM TO BUILD A SIMPLE CALCULATOR THAT CAN PERFORM ADDITION, SUBTRACTION, MULTIPLICATION, AND DIVISION OPERATIONS WITH A GUI.

2) WRITE A PROGRAM TO DEVELOP A CONVERTER TOOL THAT CAN CONVERT CURRENCIES BETWEEN DIFFERENT COUNTRIES BASED ON LIVE EXCHANGE RATES.

3) WRITE A PROGRAM TO CREATE A TIC-TAC-TOE GAME THAT CAN BE PLAYED BETWEEN TWO PLAYERS AND HAS AN OPTION TO PLAY AGAINST THE COMPUTER USING MINIMAX ALGORITHM.

4) WRITE A PROGRAM TO DESIGN A WEATHER APP THAT FETCHES REAL-TIME WEATHER DATA USING A PUBLIC API AND DISPLAYS CURRENT WEATHER, HUMIDITY, AND TEMPERATURE.

5) WRITE A PROGRAM TO BUILD A SIMPLE PERSONAL BUDGETING TOOL THAT TRACKS INCOME AND EXPENSES, CATEGORIZES SPENDING, AND GENERATES REPORTS.

6) WRITE A PROGRAM TO CREATE A PASSWORD GENERATOR THAT CAN CREATE STRONG, RANDOM PASSWORDS WITH SPECIFIED LENGTH AND CHARACTER TYPES (uppercase, lowercase, digits, symbols).

7) WRITE A PROGRAM TO DEVELOP A CONTACT MANAGEMENT SYSTEM THAT ALLOWS USERS TO ADD, EDIT, DELETE, AND SEARCH CONTACTS, AND STORE THEM LOCALLY OR IN A DATABASE.

8) WRITE A PROGRAM TO CREATE A STUDENT MANAGEMENT SYSTEM THAT STORES AND RETRIEVES STUDENT DETAILS, ALLOWS SEARCH AND UPDATE, AND GENERATES REPORT CARDS.

9) WRITE A PROGRAM TO DESIGN A SIMPLE TO-DO LIST APPLICATION THAT ALLOWS USERS TO ADD, EDIT, DELETE, AND MARK TASKS AS COMPLETE, WITH PRIORITY LEVELS AND DUE DATES.

10) WRITE A PROGRAM TO DEVELOP A CONVERTER THAT CAN CONVERT TEXT TO SPEECH (TTS) AND SPEECH TO TEXT (STT) USING MICROPHONE INPUT.

11) WRITE A PROGRAM TO IMPLEMENT A WEATHER FORECASTING SYSTEM THAT TAKES LOCATION AS INPUT AND RETURNS A 7-DAY WEATHER FORECAST, INCLUDING HUMIDITY, TEMPERATURE, AND PRECIPITATION.

12) WRITE A PROGRAM TO CREATE A SIMPLE CHAT APPLICATION THAT ALLOWS MULTIPLE USERS TO SEND AND RECEIVE MESSAGES IN REAL TIME USING SOCKET PROGRAMMING.

13) WRITE A PROGRAM TO DESIGN A SIMPLE GAME WHERE THE USER MUST GUESS A RANDOM NUMBER BETWEEN 1 AND 100, WITH THE GAME PROVIDING HINTS IF THE GUESS IS TOO HIGH OR TOO LOW.

14) WRITE A PROGRAM TO DEVELOP A FILE COMPRESSION TOOL THAT TAKES A FILE, COMPRESSES IT TO A ZIP FILE, AND EXTRACTS THE FILE WHEN NEEDED.

15) WRITE A PROGRAM TO BUILD A QUIZ APP THAT TAKES QUESTIONS AND ANSWERS FROM A DATABASE OR FILE, DISPLAYS THEM, AND CALCULATES THE FINAL SCORE.

16) WRITE A PROGRAM TO CREATE A BASIC IMAGE SLIDESHOW TOOL THAT TAKES A DIRECTORY OF IMAGES AND PRESENTS THEM IN A SLIDE SHOW WITH TRANSITIONS AND TIME INTERVALS.

17) WRITE A PROGRAM TO DESIGN A BASIC MUSIC PLAYER THAT CAN PLAY AUDIO FILES (MP3, WAV) AND SUPPORT PLAY, PAUSE, STOP, AND SKIP FUNCTIONS WITH A GUI.

18) WRITE A PROGRAM TO DEVELOP A SIMPLE CALENDAR APP THAT DISPLAYS A MONTHLY VIEW AND ALLOWS USERS TO ADD EVENTS WITH NOTIFICATIONS.

19) WRITE A PROGRAM TO CREATE A MINI BOOK LIBRARY SYSTEM THAT ALLOWS USERS TO ADD, REMOVE, SEARCH, AND DISPLAY BOOKS, AND KEEP TRACK OF BORROWED BOOKS.

20) WRITE A PROGRAM TO IMPLEMENT A SIMPLE E-COMMERCE SHOPPING CART SYSTEM WHERE USERS CAN ADD, REMOVE, AND UPDATE PRODUCTS IN THEIR CART AND CALCULATE THE TOTAL COST.

21) WRITE A PROGRAM TO BUILD A BMI CALCULATOR THAT ACCEPTS THE USER'S WEIGHT AND HEIGHT, CALCULATES THEIR BODY MASS INDEX (BMI), AND PROVIDES A DIET RECOMMENDATION BASED ON THE RESULT.

22) WRITE A PROGRAM TO DESIGN A SIMPLE FLASHCARD APP THAT ALLOWS USERS TO CREATE AND REVIEW FLASHCARDS FOR STUDYING, WITH THE OPTION TO MARK CARDS AS MASTERED.

23) WRITE A PROGRAM TO DEVELOP A SIMPLE CONTACT BOOK THAT CAN STORE NAME, EMAIL, PHONE NUMBER, AND ADDRESS FOR EACH CONTACT, AND ALLOWS SEARCH AND FILTERING.

24) WRITE A PROGRAM TO CREATE A SIMPLE SHOPPING LIST APP THAT ALLOWS USERS TO ADD, REMOVE, AND EDIT ITEMS IN A SHOPPING LIST WITH A SIMPLE TEXT-BASED OR GUI INTERFACE.

25) WRITE A PROGRAM TO BUILD A SIMPLE RECIPE APP THAT ALLOWS USERS TO INPUT, STORE, SEARCH, AND FILTER RECIPES BASED ON INGREDIENTS OR CATEGORIES.