

# COMPUTER SCIENCE

## Paper – 2

### (PRACTICAL)

(Reading Time: 15 minutes)

(Planning Session AND Examination Session: Three Hours)

---

*The total time to be spent on the Planning and the Examination Session is Three hours.*

*After completing the Planning Session, the candidate may begin with the Examination Session.*

*A maximum of 90 minutes is permitted to begin the Examination Session.*

*However, if candidates finish earlier, they are to be permitted to begin the Examination Session.*

*(Maximum Marks: 80)*

---

**As it is a practical examination the candidate is expected to do the following:**

1. Write an algorithm for the selected problem. [10]  
(Algorithm should be expressed clearly using any standard scheme such as pseudo code or in steps which are simple enough to be obviously computable.)
  2. Write a program in **JAVA** language. The program should follow the algorithm and should be logically and syntactically correct. [20]
  3. Document the program using mnemonic names / comments, identifying and clearly describing the choice of data types and meaning of variables. [10]
  4. Code / Type the program on the computer and get a printout (hard copy). Typically, this should be a program that compiles and runs correctly. [10]
  5. Test run the program on the computer using the given sample data and get a printout of the output in the format specified in the problem. [20]
  6. Viva-Voce on the **Selected Problem.** [20]
-

### Question 1

A bank intends to design a program to display the denomination of an input amount, up to 5 digits. The available denomination with the bank are of rupees 1000 , 500 , 100 , 50 , 20 , 10 , 5 , 2 , and 1.

Design a program to accept the amount from the user and display the break-up in descending order of denomination. (i.e. preference should be given to the highest denomination available) along with the total number of notes. [Note: Only the denomination used should be displayed]. Also print the amount in words according to the digits.

#### Example 1

INPUT : 14856  
OUTPUT : ONE FOUR EIGHT FIVE SIX

DENOMINATION	:	1000	x	14	=	14000
		500	x	1	=	500
		100	x	3	=	300
		50	x	1	=	50
		5	x	1	=	5
		1	x	1	=	1
TOTAL					=	14856
TOTAL NUMBER OF NOTES					=	21

#### Example 2

INPUT : 6043  
OUTPUT : SIX ZERO FOUR THREE

DENOMINATION	:	1000	x	6	=	6000
		20	x	2	=	40
		2	x	1	=	2
		1	x	1	=	1
TOTAL					=	6043
TOTAL NUMBER OF NOTES					=	10

#### Example 3

INPUT : 235001  
OUTPUT : INVALID AMOUNT

## Question 2

A positive whole number 'n' that has 'd' number of digits is squared and split into two pieces, a right-hand piece that has 'd' digits and a left-hand piece that has remaining 'd' or 'd-1' digits. If the sum of the two pieces is equal to the number, then 'n' is a Kaprekar number. The first few Kaprekar numbers are: 9, 45, 297 .....

### Example 1:

9

$9^2 = 81$ , right-hand piece of 81 = 1 and left hand piece of 81 = 8

Sum = 1 + 8 = 9, i.e. equal to the number.

### Example 2:

45

$45^2 = 2025$ , right-hand piece of 2025 = 25 and left hand piece of 2025 = 20

Sum = 25 + 20 = 45, i.e. equal to the number.

### Example 3:

297

$297^2 = 88209$ , right-hand piece of 88209 = 209 and left hand piece of 88209 = 88

Sum = 209 + 88 = 297, i.e. equal to the number.

Given the two positive integers  $p$  and  $q$ , where  $p < q$ , write a program to determine how many Kaprekar numbers are there in the range between  $p$  and  $q$  (both inclusive) and output them.

The input contains two positive integers  $p$  and  $q$ . Assume  $p < 5000$  and  $q < 5000$ . You are to output the number of Kaprekar numbers in the specified range along with their values in the format specified below:

### SAMPLE DATA:

INPUT:

$p = 1$

$q = 1000$

OUTPUT:

THE KAPREKAR NUMBERS ARE:-

1, 9, 45, 55, 99, 297, 703, 999

FREQUENCY OF KAPREKAR NUMBERS IS: 8

### Question 3

Input a paragraph containing 'n' number of sentences where  $(1 \leq n \leq 4)$ . The words are to be separated with a single blank space and are in UPPERCASE. A sentence may be terminated either with a full stop '.' or a question mark '?' only. Any other character may be ignored. Perform the following operations:

- i. Accept the number of sentences. If the number of sentences exceeds the limit, an appropriate error message must be displayed.
- ii. Find the number of words in the whole paragraph
- iii. Display the words in ascending order of their frequency. Words with same frequency may appear in any order.

#### Example 1

INPUT:      Enter number of sentences.  
              1  
              Enter sentences.  
              TO BE OR NOT TO BE.

OUTPUT:    Total number of words: 6

WORD	FREQUENCY
OR	1
NOT	1
TO	2
BE	2

#### Example 2

INPUT:      Enter number of sentences  
              3  
              Enter sentences.  
              THIS IS A STRING PROGRAM.IS THIS EASY?YES,IT IS.

OUTPUT:    Total number of words: 11

WORD	FREQUENCY
A	1
STRING	1
PROGRAM	1
EASY	1
YES	1
IT	1
THIS	2
IS	3

#### Example 3

INPUT:      Enter number of sentences  
              5  
OUTPUT:    Invalid entry