ISC YEAR 2025

COMPUTER SCIENCE - PAPER 2

PRACTICAL EXAMINATION

Instructions and Guidelines for Visiting Examiners

1. Preparation of Question Paper/s and conduct of the Practical Examination
2. Blueprint of the Question Paper
3. Question Specific Guidelines & Instructions
4. Guidelines for Evaluation of Computer Science-Paper 2(Practical)
5. Sample Question Papers

I. PREPARATION OF QUESTION PAPER/S AND CONDUCT OF PRACTICAL EXAMINATION

The Practical Examination must be conducted between 1st October 2024 and 31st January 2025.

Steps to be followed for conducting Computer Science Practical Examination.

Deciding the number of Batches and Dates for the Practical Examination/s

1. The Visiting Examiner must collect the soft copies/hard copies of all documents related to the conduct of the Practical Examination from the Head of the School.
2. The Head of the School, in consultation with the Visiting Examiner and the Chief/Supervising Examiner should decide the number of batches in which the candidates of the School would take the practical examination. This should be done keeping in mind the size of the laboratory and total number of candidates appearing for Computer Science practical examination.
3. Each batch should be assigned a batch number – e.g., Batch 1 (Index no. 225001/001 to 225001/010); Batch 2 (Index no. 225001/011 to 225001/020). A record of this information should be maintained by the Head of the School. A copy of this record should be submitted to the Convener.
4. Dates of Practical Examination for each batch must be decided according to the convenience of both the Visiting Examiner and the School.
5. The candidates must be informed of the dates of the Practical Examination (for their batch) ***at least one week in advance***.

Finalising the Question Papers/s

1. The Visiting Examiner will be required to set the question paper/s for the candidate(s) concerned based on the syllabus for the ISC Year 2025 Practical Examination and other guidelines provided by CISCE. The question paper/s prepared could be similar to the ones in the Sample Question Papers provided by CISCE inthis document.
2. The Visiting Examiner should select the questions, as per the Blueprint and other guidelines provided by CISCE, include relevant details and run the programs where required, to ensure that the desired results are obtained before finalising the Question Paper.
3. The Visiting Examiner should use a range of questions (as provided in the sample questions) for each batch, while ensuring that all batches get Question Papers of similar difficulty level.
4. After finalising the Question Paper/s, the Visiting Examiner must inform the Head of the School, regarding the arrangements to be made in the Computer laboratory for the conduct of the practical examination.

Preparation and Storage of Question Paper/s prepared

5. The Head of the School, in presence of the Visiting Examiner and the Chief/Supervising Examiner, should ensure that the question papers are photocopied in adequate numbers (as per the number of candidates concerned), well in advance.

6. The photocopied question paper/s should be packed and sealed in envelope(s) by the Visiting Examiner which should be labelled as follows:

Subject: Computer Science-Paper 2 Practical Examination

Date of Examination: ______________________

Batch Number: ___________________________

Index No.(s):____________________

Number of Question Papers packed:____________

Both the Chief/Supervising Examiner, and the Visiting Examiner should put their signatures across the seal of the envelope(s), with the date.

7. The Visiting Examiner should handover the sealed envelope(s) containing copies of the finalised question paper/s to the Chief/Supervising Examiner for safe custody.

8. The Chief/Supervising Examiner must ensure that the sealed envelope(s) of the question papers are kept securely, under lock and key, in a cupboard, in the office of the Head of the School.

9. All concerned must maintain strict confidentiality of the finalised question papers.

Uploading of Marks awarded by the Visiting Examiner.

10. The marks awarded by the Visiting Examiner, as per the Guidelines for Evaluation of Computer Science Paper 2 (Practical) (given on page 6), must be uploaded onto the CAREERS Portal after the evaluation of answer script(s) of candidate(s)concerned.

11. The Visiting Examiner must also evaluate and upload the marks awarded for the Project work and Practical files of the candidates.

12. After uploading the marks, the data must be saved on the Portal by clicking the 'SAVE' button. The final submission of marks to CISCE should then be made after the marks of all the candidates of the School, for the given subject, have been uploaded onto the Portal.

13. It is necessary to ensure that the marks of all candidates are entered correctly, before submitting the same to CISCE.

14. After the final submission of marks to CISCE, the Visiting Examiner must take a print-out of the marks submitted, and put his name, signature and date on each sheet. This print-out of marks must be handed over in a sealed envelope to the Head of the School.

15. The marks awarded to the candidates must be treated as strictly confidential and must not be disclosed to anyone by the Visiting Examiner.

16. The evaluated answer scripts must be packed securely in durable envelope(s). The Chief/Supervising Examiner must sign the packed answer script envelopes to ensure that the answer scripts were packed in her/his presence and that she/he has verified the number of scripts enclosed in the envelope.

    All answer script envelopes must be addressed to:

    The Chief Executive & Secretary
    Council for the Indian School Certificate Examinations
    Plot No. 35-36, Sector 6,
    Pushp Vihar, Saket
    New Delhi – 110017

BLUEPRINT OF THE PRACTICAL QUESTIONPAPER

The Computer Science Practical Question Paper consists of Three questions and the candidate must select any ONE question. Each question must consist of four components. The details /marks distribution is given below:

1.	Algorithm	**3 marks**
2.	Program in Java with documentation	**7 marks**
3.	Hard copy / printout	**2marks**
4.	Output / execution	**3 marks**
	Total	**15 Marks**

The three programs must be based on the topics as given below:

Question 1.	Number Logic / Date concept / Time concept/ Number encryption
Question2.	Single dimensional array / Double dimensional array
Question 3.	String manipulation / String encryption

In addition to the above, the Visiting Examiner will also be required to evaluate the following for each candidate:

• Programming assignments done throughout the year (by the teacher)	**10 marks**
• Programming assignments done throughout the year (by the V.E.)	**5 marks**
Total	**15 Marks**

QUESTION SPECIFIC GUIDELINES & INSTRUCTIONS

The Visiting Examiner, in consultation with the subject teacher, must decide on the programs to set the question paper. The Sample Question Papers provided by CISCE may be referred to for this purpose.

The questions must be set as follows:

Question 1: Number Logic / Date concept / Time Concept

The program should check the validity of a given input / range and output the desired result in a formatted manner. Logic for checking the validity, extraction of digits from a number, changing a numeric literal to String literal and vice-versa may be tested.

- Number logic- to extract digits from a number, checking and generating a particular type of number (i.e. Smith Number, Kaprekar Number etc.).
- Date concept- to check validity of dates, to find the number of days between two given dates in the same calendar year, future date etc.
- Time concept- to find the time elapsed between any two-given time in a particular day, to find the future time after some duration (in hours and minutes)
- Number encryption – to convert decimal numbers to different bases (Binary, Octal and Hexadecimal) and vice versa. Coding of numbers using their ASCII values

Question 2: Single dimensional array/Double dimensional array

The program should check the validity of the size of the matrix/array and output the desired result in a formatted manner. Formula to check if the elements lie on the boundary, non-boundary, diagonal etc. in a given matrix/array.

- Single dimensional array- merging of two arrays, to arrange the array elements in a particular manner etc.
- Double dimensional array / matrix- to find the sum/sort the elements of each row/column, boundary elements, non-boundary elements, saddle point etc.
- Square matrix to find sum/sort elements of the diagonal, maximum and minimum value with location in the matrix, mirror/inversion of the matrix etc.

Question 3: String manipulation / String encryption

The program to check the validity of a sentence/paragraph with the terminating character given. Logic to extract the words from a sentence, characters from words, vowels, and consonants etc.

- Paragraph input with a maximum of two sentences and finding the number of words in each sentence.
- To separate the words beginning with a vowel in a sentence and find its frequency.
- To change the Upper-case characters to lower case and vice versa in each sentence along with the count of palindrome words present in the sentence, if any. Encoding and decoding of characters, sorting of characters in a string etc.

IV.GUIDELINES FOR EVALUATION OF COMPUTER SCIENCE PAPER–2 (PRACTICAL)

(for the use of the Visiting Examiners)

Marks (Out of a total of 30) for this paper are to be distributed as follows:		
A.	Assessment of practical examination:	**15 Marks**
B.	Assessment of programming done throughout the year:	**15 Marks**
Part A: Assessment of Practical Examination (15 Marks)		
Candidates are expected to plan their programs and test run them on the computer. The various stages are to be assessed as follows:		
1.	**Algorithm:**	**[3]**
	❖ Choice of algorithm and implementation strategy.	[1½]
	❖ Complete and clearly expressed algorithm using any standard schemes (i.e. pseudo code or stepwise) and according to the program.	[1½]
2. (a)	**Java Program: (Selected by candidate)**	**[5]**
	❖ Knowledge of Input requirement and the choice of data type.	[1]
	❖ Clear description of classes, functional behavior, functional argument and return type and any formulae or special logic/method used.	[1]
	❖ The program follows the algorithm correctly and is logically correct.	[2]
	❖ Use of correct and formatted output statement.	[1]
(b)	**Documentation** to be provided in the handwritten or printed coded program by comments/mnemonic names.	**[2]**
3.	**The final listing (hard copy)** that the candidate submits at the end, follows the algorithm and is logically correct. Large differences between planned program and the printout will result in loss of marks.	**[2]**
4.	**Execution and testing**:	**[3]**
	❖ The program runs successfully on the sample inputs to produce the correct sample outputs (correct output on known inputs).	[2]
	❖ The examiner should ask the candidate to run program on other inputs to ensure that it is correct. These would typically be the extreme boundary conditions (correct output on unknown inputs).	[1]
	❖ Candidates are required to execute programs using JDK environment/BlueJ/Eclipse/NetBeans. *(BlueJ allows the user to write a class, create objects and test their functionality without having to write public static void main ( ). Hence, the program executed using BlueJ environment should also be accepted.)*	

Part B : Assessment of programming done throughout the year (15 Marks)

The Subject Teacher will assess the year's work of candidates (assignments done as practical work throughout the year) and award marks out of 10. These marks, along with the year's work, should be made available to the Visiting Examiner. After taking the internally awarded marks into consideration, the Visiting Examiner should award marks out of **5**, for the year's work of candidates.The total marks for continuous evaluation, therefore, shall be out of **15** marks (out of 10 marks awarded through internal evaluation and out of 5 marks awarded by the Visiting Examiner).

ISC YEAR 2025

COMPUTER SCIENCE PAPER 2

PRACTICAL EXAMINATION

SAMPLE PAPERS

PREPARING THE QUESTION PAPER

- The **Sample Question Papers** provided by CISCE include different questions for each type of Practical Work given in the syllabus. These questions may be used by the Visiting Examiner for setting the question Paper/s.
- The Visiting Examiners are advised to go through the ***Instructions and Guidelines for Visiting Examiners***, before setting the Question Paper/s.
- The questions must be selected as per the Blueprint and other details provided by CISCE, include relevant details and run the program himself/herself, where required, to ensure that the desired results are obtained before finalizing the Question Paper.
- The text for the questions may be similar to the text used for the Sample Questions.
- The provided **Top Sheet,** detailing the time provided for reading the Question Paper, the duration of the Examination and other necessary information, must be attached as the ***first sheet*** of the Question Paper.

COMPUTER SCIENCE

PAPER – 2

PRACTICALS

(Maximum Marks: 30)

Time allowed: Three Hours

(Candidates are allowed additional 15 minutes for only reading the paper.~~)~~
They must NOT start writing during this time.)

The total time to be spent on the Planning Session and the Examination Session is Three hours.
After completing the Planning Session, the candidate may begin the Examination Session.
A maximum of 90 minutes is permitted for the Planning Session.
However, if candidates finish earlier, they are permitted to begin the Examination Session.

---

*This paper consists of **three** problems from which candidates are required to attempt **any one** problem.*

Candidates are expected to do the following:

A. Planning Session:

1. Write an algorithm for the selected problem. **[3marks]**

   (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. Document the program using mnemonic names / comments, identifying and clearly describing the choice of data types and meaning of variables. **[7marks]**

B. Examination Session:

1. Code / Type the program on the computer and get a printout (hard copy). Typically, this should be a program that compiles and runs correctly. **[2marks]**

2. 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. **[3marks]**

Note: *The candidates must not carry any stationery, items such as pen / pencil / eraser to the Computer Laboratory for the Examination Session.*

Top Sheet: To be attached as the first page of the Question Paper.

SAMPLE PAPER 1

Solve **any one** of the following Problems.

Question 1

A number is said to be a **Goldbach** number, if the number can be expressed as the addition of two odd prime number pairs. If we follow the above condition, then we can find that every even number larger than 4 is a **Goldbach** number because it must have any pair of odd prime number pairs.

Example: 6 = 3,3 (ONE PAIR OF ODD PRIME)
10 = 3,7 and 5 , 5 (TWO PAIRS OF ODD PRIME)

Write a program to enter any positive EVEN natural number 'N' where (1<=N<=50) and generate odd prime twin of 'N'

Test your program for the following data and some random data.

Example 1

INPUT: N = 14

OUTPUT: ODD PRIME PAIRS ARE: 3, 11
7, 7

Example 2

INPUT: N = 20

OUTPUT: ODD PRIME PAIRS ARE: 17, 3
13, 7

Example 3

INPUT: N = 44

OUTPUT: ODD PRIME PAIRS ARE: 41, 3
37, 7
31, 13

Example 4

INPUT: N = 25

OUTPUT: INVALID INPUT

Question 2

Write a program to declare a matrix A [ ] [ ] of order (M × N) where ‘M’ is the number of rows and ‘N’ is the number of columns such that both M and N must be greater than 2 and less than10. Allow the user to input integers into this matrix. Display appropriate error message for an invalid input.

Perform the following tasks on the matrix.

(a) Display the input matrix

(b) Shift each row one step upwards so the first row will become the last row 2nd row will be the 1st row and so on

(c) Display the rotated matrix along with the highest element and its location in the matrix

Test your program for the following data and some random data:

Example 1

INPUT:

```
M =3
N = 4
Enter elements in the matrix:
100    90    87    76
200   500   167   998
 77   567    89   254
```

OUTPUT:

```
FORMED MATRIX AFTER ROTATING:
200   500   167   998
 77   567    89   254
100    90    87    76
Highest element:  998   ( Row: 0   and   Column: 3 )
```

Example 2

INPUT:

```
M =4
N = 3
Enter elements in the matrix:
 54    120    187
 78     55    289
134     67     89
 63    341    122
```

OUTPUT:

```
FORMED MATRIX AFTER ROTATING:
 78     55    289
134     67     89
 63    341    122
 54    120    187
Highest element:  341   ( Row: 2   and   Column: 1 )
```

Example 3

INPUT:

```
M = 2
N =  3
```

OUTPUT:

```
SIZE IS OUT OF RANGE.  INVALID ENTRY
```

Question 3

Write a program to accept a sentence which may be terminated by either **'.'** ,**'?'**or **'!'** only. The words may be separated by a single blank space and should be case-insensitive.

Perform the following tasks:

(a) Determine if the accepted sentence is a Pangram or not.
[A Pangram is a sentence that contains every letter of the alphabet at least once.]
Example: "The quick brown fox jumps over the lazy dog"

(b) Display the first occurring longest and shortest word in the accepted sentence.

Test your program for the following data and some random data:

Example 1

INPUT: Pack my box with five dozen liquor jugs.

OUTPUT: IT IS A PANGRAM
LONGEST WORD: dozen
SHORTEST WORD: my

Example 2

INPUT: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.

OUTPUT: IT IS A PANGRAM
LONGEST WORD: QUICK
SHORTEST WORD: THE

Example 3

INPUT: Hello my World.

OUTPUT: IT IS NOT A PANGRAM
LONGEST WORD: Hello
SHORTEST WORD: my

Example 4

INPUT: Alas ! it failed #

OUTPUT: INVALID INPUT

SAMPLE PAPER 2

Solve **any one** of the following Problems.

Question 1

Write a program in JAVA to accept day number (between 1 and 366) and year (yyyy) from the user and display the corresponding date. Also accept 'N' from the user where (1<=N<=100) to compute and display the future date 'N' days after the given date. Display error message if the value of the day number or 'N' are not within the limit. Day number is calculated taking 1st January of the given year as 1.

Test your program with given set of data and some random data

Example 1

INPUT: DAY NUMBER: 50

YEAR: 2024

N: 25

OUTPUT: ENTERED DATE: FEBRUARY 19, 2024

25 DAYS LATER: MARCH 15, 2024

Example 2

INPUT: DAY NUMBER: 321

YEAR: 2024

N: 77

OUTPUT: ENTERED DATE: NOVEMBER 16, 2024

77 DAYS LATER: FEBRUARY 1, 2025

Example 3

INPUT: DAY NUMBER: 400

YEAR: 2024

N: 125

OUTPUT: INCORRECT DAY NUMBER

INCORRECT VALUE OF 'N'

Question 2

Write a program to declare a square matrix A[ ][ ] of order M × M where `M' is the number of rows and the number of columns, such that M must be greater than 2 and less than 10. Accept the value of M as user input. Display an appropriate message for an invalid input. Allow the user to input integers into this matrix. Perform the following tasks:

(a) Display the original matrix.

(b) Check if the given matrix is Symmetric or not.

A square matrix is said to be Symmetric, if the element of the i^{th} row and j^{th} column is equal to the element of the j^{th} row and i^{th} column.

(c) Find the sum of the elements of left diagonal and the sum of the elements of right diagonal of the matrix and display them.

Test your program for the following data and some random data:

Example 1

```
INPUT:     M = 3
           Enter elements of the matrix:
           1     2     3
           2     4     5
           3     5     6

OUTPUT:    ORIGINAL MATRIX
           1     2     3
           2     4     5
           3     5     6

           THE GIVEN MATRIX IS SYMMETRIC
           The sum of the left diagonal = 11
           The sum of the right diagonal = 10
```

Example 2

```
INPUT:     M = 4
           Enter elements of the matrix:
           7     8     9     2
           4     5     6     3
           8     5     3     1
           7     6     4     2

OUTPUT:    ORIGINAL MATRIX
           7     8     9     2
           4     5     6     3
           8     5     3     1
           7     6     4     2

           THE GIVEN MATRIX IS NOT SYMMETRIC
           The sum of the left diagonal = 17
           The sum of the right diagonal = 20
```

Example 3

```
INPUT:     M = 12

OUTPUT:    SIZE IS OUT OF RANGE
```

Question 3

Most (NOT ALL) cell phone keypads look like the following arrangement (the letters are above the respective number)

1	ABC 2	DEF 3
GHI 4	JKL 5	MNO 6
PQRS 7	TUV 8	WXYZ 9
	[SPACE] 0	

For sending text / SMS the common problem is the number of keystrokes to type a particular text.

For example, the word "STOP", there are a total of 9 keystrokes needed to type the word. You need to press the key 7 four times, the key 8 once, the key 6 three times and the key 7 once to get it.

Develop a program code to find the number of keystrokes needed to type the text.

For this problem, accept just one word without any punctuation marks, numbers or white spaces and the text message would consist of just 1 word.

Test your data with the sample data and some random data :

Example 1:

INPUT: DEAR

OUTPUT: Number of keystrokes = 7

Example 2:

INPUT: Thanks

OUTPUT: Number of keystrokes = 12

Example 3:

INPUT: Good-Bye

OUTPUT: INVALID ENTRY

SAMPLE PAPER 3

Solve **any one** of the following Problems.

Question 1

A unique-digit integer is a positive integer (without leading zeros) with no duplicate digits. For example, 7, 135, 214 are all unique-digit integers whereas 33, 3121, 300 are not.

Given two positive integers *m* and *n*, where *m* < *n*, write a program to determine how many unique-digit integers are there in the range between *m* and *n* (both inclusive) and output them.

The input contains two positive integers *m* and *n*. Assume *m* < *30000* and *n* < *30000*. You are to output the number of unique-digit integers in the specified range along with their values in the format specified below:

Test your program for the following data and some random data.

Example 1

```
INPUT:     m = 100
           n =  120

OUTPUT:  THE UNIQUE-DIGIT INTEGERS ARE:
          102, 103, 104, 105, 106, 107, 108, 109, 120

          FREQUENCY OF UNIQUE-DIGIT INTEGERS IS: 9
```

Example 2

```
INPUT:     m = 2505
           n =  2525

OUTPUT:  THE UNIQUE-DIGIT INTEGERS ARE:

          2506, 2507, 2508, 2509, 2510, 2513, 2514, 2516, 2517, 2518, 2519

          FREQUENCY OF UNIQUE-DIGIT INTEGERS IS: 11
```

Example 3

```
INPUT:     m = 2520
           n =  2529

OUTPUT:  THE UNIQUE-DIGIT INTEGERS ARE: NIL

         FREQUENCY OF UNIQUE-DIGIT INTEGERS IS: 0.
```

Question 2

Write a program to declare a matrix A [ ] [ ] of order (M x N) where ‘M’ is the number of rows and ‘N’ is the number of columns such that both M and N must be greater than 2 and less than 20. Allow the user to input integers into this matrix. Perform the following tasks on the matrix:

Perform the following tasks on the matrix.

(a) Display the input matrix.

(b) Find the maximum and minimum value in the matrix and display them along with their position.

(c) Sort the elements of the matrix in descending order using any standard sorting technique and rearrange them in the matrix.

(d) Output the rearranged matrix.

Test your program for the following data and some random data:
Example 1

```
INPUT:     M = 3
           N = 4
           Enter elements of the matrix:
           8     7     9     3
           -2    0     4     5
           1     3     6     -4

OUTPUT:    ORIGINAL MATRIX

           8     7     9     3
           -2    0     4     5
           1     3     6     -4

           LARGEST  NUMBER:   9
           ROW = 0
           COLUMN = 2

            SMALLEST NUMBER:   -4
           ROW = 2
           COLUMN = 3

           REARRANGED MATRIX

           -4    -2    0     1
            3     3    4     5
            6     7    8     9
```

Example 2

```
INPUT:      M = 3
            N = 3
             Enter elements of the matrix:
             7      9      3
             -2     4      5
             1      16     4

OUTPUT:    ORIGINAL MATRIX

             7      9      3
             -2     4      5
             1      16     4

            LARGEST  NUMBER:   16
            ROW = 2
            COLUMN = 1

             SMALLEST NUMBER:   -2
            ROW = 1
            COLUMN = 0

            REARRANGED MATRIX

             -2     1      3
              4     4      5
              7     9      16
```

Example 3

```
INPUT:      M = 3
            N = 22

OUTPUT:     SIZE OUT OF RANGE
```

Question 3

Write a program to check if a given string is an Anagram of another string. Two strings are anagrams if they can be rearranged to form the same string. For example, "listen" and "silent" are anagrams.

Accept two strings from the user and check if they are anagrams of each other. Ensure that the comparison is case-insensitive and ignores spaces. Display an appropriate message based on whether they are anagrams or not. If any of the strings contain invalid characters (e.g., numbers or special characters), generate an error message.

Test your program with the following data and some random data:

Example 1

INPUT: Enter first string: Listen
Enter second string: Silent

OUTPUT: STRINGS ARE ANAGRAMS

Example 2

INPUT: Enter first string: Dormitory
Enter second string: Dirty room

OUTPUT: STRINGS ARE ANAGRAMS

Example 3

INPUT: Enter first string: Hello
Enter second string: World

OUTPUT: STRINGS ARE NOT ANAGRAMS

Example 4

INPUT: Enter first string: Test123
Enter second string: 321tset

OUTPUT: INVALID CHARACTERS IN STRING. INVALID INPUT