KARISHMA KUMARI

stu68b04bf399a691756384243@aicte-india.org

Personal Details

19 January 2002

9818691915

East Delhi

Test Details

Make My Trip Practice Assessment

28-08-2025

06:44 PM

Proctoring Details

IP Restricted Attempts : **0**

Code of Conduct : **Sincere**

Test score Recommendation

Recommendation

Not Recommended

Test	Score
Proctoring Score	10 / 20
Cut-off Score	28 / 80
Total Score	38 / 100

Test Score:

14%

Score:14 / 100

Candidate is highest Scorer

Ranking

27th

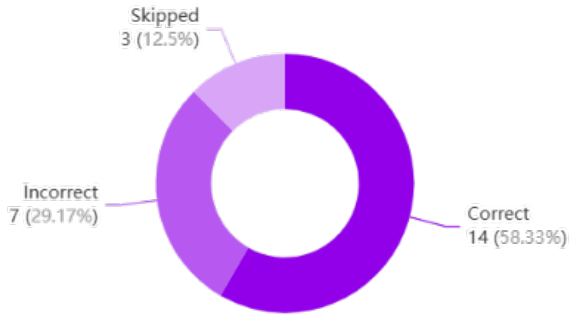
out of 161 candidates

	Score
Average Score	12
Top Score	96

Candidate Score

14

Candidate took 37 mins and 20 secs to answer 24 questions.



Easy

1. Correct Answers	2
2. InCorrect Answers	1
3. Unattempted Questions	1

Intermediate

1. Correct Answers	10
2. InCorrect Answers	5
3. Unattempted Questions	1

Hard

1. Correct Answers	2
2. InCorrect Answers	1
3. Unattempted Questions	1

Section Performance:

Technical - 1:

Marks **14/20**

Coding - 1:

Marks **0/80**

Submitted-Answers:

Technical - 1 - MCQ

Question 1:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **28 secs**

The system/370 assembler language

- [x] allows the programmer to write base registers & displacements' explicitly in the source program
- [ ] Is used to remember which of the general-purpose registers are currently available as base registers, and what base addresses they contain
- [ ] Allows multiple programs to reside in separate areas of core at the same tune
- [ ] Is a term that refers to the control programs of an operating system
- [ ] None of the above

Candidate Answer:

- [x] allows the programmer to write base registers & displacements' explicitly in the source program

Question 2:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **18 secs**

The higher versions of the operating systems are so written that program designed for earlier versions can still be run. What is it called?

- [ ] Upgradability
- [ ] Upward mobility
- [ ] Universality
- [x] Upward compatibility
- [ ] None of the above

Candidate Answer:

- [x] Upward compatibility

Question 3:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **28 secs**

The primary job of the operating system of a computer is to

- [ ] command resources
- [x] manage resources

provide utilities

be user friendly

none of the above

Candidate Answer:

manage resources

Question 4:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/1**
Time spent: **32 secs**

The Register-to-Register

have both their operands in the main store

Which perform an operation on a register operand and an operand which is located in the main store, generally leaving the result in the register, expect in the case of store operation when it is also written into the specified storage locations

Which perform indicated operations on two fast registers of the machine and have the result in one of the registers

All of the above

None of the above

Candidate Answer:

Which perform indicated operations on two fast registers of the machine and have the result in one of the registers

Question 5:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/1**
Time spent: **20 secs**

A flowchart that uses predefined symbols to describe data flow in a system is known as

program flowchart

System flowchart

Data flow diagram

All of the above

None of the above

Candidate Answer:

Candidate Answer:

Data flow diagram

Question 6:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **1 min, 31 secs**

Under multi programming, turnaround time for short jobs is usually ....and that for long jobs is slightly.....

- lengthened; shortened
- shortened; lengthened
- shortened; shortened
- shortened; unchanged
- none of the above

Candidate Answer:

shortened; lengthened

Question 7:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **20 secs**

The technique, for sharing the time of a computer among several jobs. Which switches jobs so rapidly such that each job appears to have the computer to itself:

- time sharing
- timeout
- time domain
- FIFO
- None of the above

Candidate Answer:

time sharing

Question 8:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **27 secs**

Let LASTPST, LASTIN, LASTPRE denote the last vertex visited in a postorder, inorder and preorder traversal respectively of a complete binary tree. Which of the following is always true

- LASTIN = LASTPOST
- LASTIN = LASTPRE

- LASTPRE = LASTPOST
- None of the above

Candidate Answer:

- LASTIN = LASTPRE

Question 9:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **1 min, 44 secs**

The maximum number of nodes in a binary tree of height k is

- 2^(k+1) -1, k "e 1
- 2^(k-1), k >= 1
- 2^(k+1), k "e1
- 2^(k+1), k >= 1

Candidate Answer:

- 2^(k+1) -1, k "e 1

Question 10:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/1**
Time spent: **55 secs**

Which one of the following is false about a strictly binary tree

- A binary tree is called strictly binary tree, if every non -leaf node of it has non-empty left and right sub tree
- A complete binary tree of depth d is the strictly binary tree all of whose leaves are at the level d
- A strictly binary tree with n leaves always contact 2n - 1 nodes
- In the nodes of strictly binary tree of depth d must be at the level d

Candidate Answer:

- A complete binary tree of depth d is the strictly binary tree all of whose leaves are at the level d

Question 11:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **1 min, 32 secs**

If a Queue of capacity n is implemented using an array of `size n+1` and is initialized with REAR = FRONT = 0. Then what should be the condition to detect the full and empty queue.

- Full: (REAR+1) % (n+1) == FRONT, Empty: REAR == FRONT

Full: REAR % (n+1) == FRONT, Empty: REAR == FRONT

Full: (REAR+1) % n == FRONT, Empty: REAR == FRONT

Full: REAR % n == FRONT, Empty: (REAR+1) == FRONT

Candidate Answer:

Full: (REAR+1) % (n+1) == FRONT, Empty: REAR == FRONT

Question 12:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/1**
Time spent: **36 secs**

What will be the output of given function? Assume stack has been implemented and has infinite capacity.

```
void func(int n)
{
  Stack S;
  while(n>0)
  {
    push(&S,n%2);
    n=n/2;
  }
  while(!isEmpty(&S)
  {
    printf("%d",pop(&S));
  }
}
```

Binary representation of n

Binary representation of n in reverse order

Octal representation of n

None

Candidate Answer:

Binary representation of n in reverse order

Question 13:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **27 secs**

Which of the following traversal outputs the data in sorted order in a BST?

Preorder

Inorder

Postorder

Level order

Candidate Answer:

Inorder

Question 14:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **15 secs**

Which traversal of tree resembles the breadth first search of the graph?

Preorder

Inorder

Post order

Level order

Candidate Answer:

Level order

Question 15:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **35 secs**

What is the worst case time complexity for search, insert and delete operations in a general Binary Search Tree?

O(n) for all

O(Logn) for all

O(Logn) for search and insert, and O(n) for delete

O(Logn) for search, and O(n) for insert and delete

Candidate Answer:

O(n) for all

Question 16:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/1**
Time spent: **1 min, 25 secs**

Consider the following code snippet in C. The function print() receives root of a Binary Search Tree (BST) and a positive integer k as arguments.

```
// A BST node
struct node {
    int data;
```

```
    struct node *left, *right;
};

int count = 0;

void print(struct node *root, int k)
{
    if (root != NULL && count <= k)
    {
        print(root->right, k);
        count++;
        if (count == k)
          printf("%d ", root->data);
       print(root->left, k);
    }
}
```

What is the output of print(root, 3) where root represent root of the following BST.

- 10
- 16
- 20
- 20 10

Candidate Answer:

- 20 10

Question 17:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/1**
Time spent: **35 secs**

Which of the following is a true about Binary Trees

- Every binary tree is either complete or full.
- Every complete binary tree is also a full binary tree
- Every full binary tree is also a complete binary tree.
- No binary tree is both complete and full.
- None of the above

Candidate Answer:

✗ Every full binary tree is also a complete binary tree.

Question 18:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **27 secs**

A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and I = 10, what is the value of n?

- 6
- 3
- 4
- ✓ 5

Candidate Answer:

- ✓ 5

Question 19:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **32 secs**

In a complete k-ary tree, every internal node has exactly k children or no child. The number of leaves in such a tree with n internal nodes is:

- nk
- (n – 1) k+ 1
- ✓ n(k – 1) + 1
- n(k – 1)

Candidate Answer:

- ✓ n(k – 1) + 1

Question 20:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **1/1**
Time spent: **38 secs**

What does the following function do for a given binary tree?

```
int fun(struct node *root)
{
   if (root == NULL)
      return 0;
   if (root->left == NULL && root->right == NULL)
      return 0;
   return 1 + fun(root->left) + fun(root->right);
}
```

- Counts leaf nodes
- ✔ Counts internal nodes
- Returns height where height is defined as number of edges on the path from root to deepest node
- Return diameter where diameter is number of edges on the longest path between any two nodes.

Candidate Answer:

- ✔ Counts internal nodes

Coding - 1 - Coding

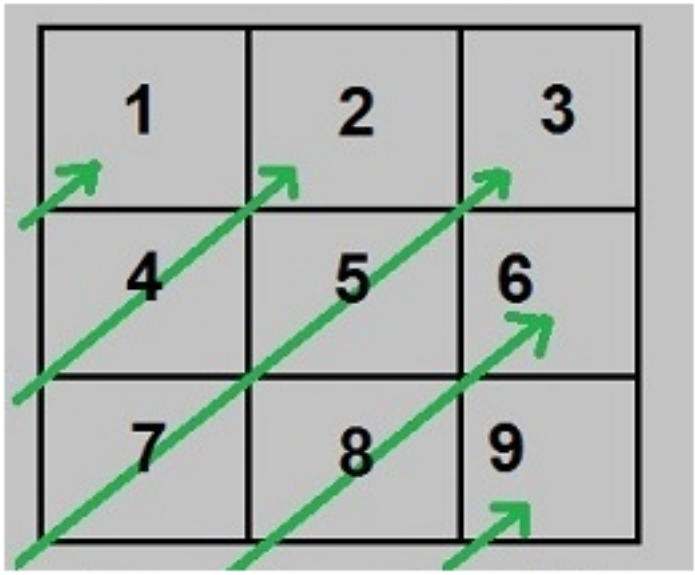
Question 1:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/20**
Time spent: **0 sec**

Ladder Matrix Traversal

Given an integer matrix, R x C, traverse it in the order as shown below:



Input:

```
* One line containing two integers R and C representing the dimensions of the matrix M as rows R, and co
* R Lines, each containing C space separated number of integers which collectively form the matrix data
```

Output:

```
* Single or multiple line containing integers without space, representing the desired traversal.
```

Contraints:

```
* 0<R<500
* 0<C<500
```

Candidate Answer:

Not Attempted

Question 2:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/20**
Time spent: **0 sec**

Numerical Wrapper

Analyze the output given in the samples for an input, N. Write a program that prints the similar pattern.

Input:

```
* One line containing an integer, N
```

Output:

```
*A pattern of numerical border around digit '1', to be deduced from samples
```

Constraints:

```
* 0 < N < 999
```

Sample Input

2

Sample Output

2 2 2
2 1 2
2 2 2

Sample Input

3

Sample Output

3 3 3 3 3
3 2 2 2 3
3 2 1 2 3
3 2 2 2 3
3 3 3 3 3

Candidate Answer:

Not Attempted

Question 3:

Total Time Spent Outside: **8 secs**
Total Move Count: **2**

Score: **0/20**
Time spent: **23 mins, 15 secs**

ROMAN NUMERALS

Write a program to convert a given string representing an integer in Roman Numeral Format to an integer in Decimal Numeral Format.

Input:

```
String S representing the roman representation of a positive integer.
```

Output:

```
Single line containing the integer.
```

Constraints:

```
The roman representation for the input will lie between 1 to 500 (both inclusive)
```

Reference:

NOTE: If the reference image is not visible, please download it from here:

`https://storage.googleapis.com/myanatomy-main/questionImages/e2455ff3-588f-4ff3-bb74-89ad4512e618_USER-RecruiterCopy_ITEM-Roman%20Numerals.gif`

Roman numerals 1-500

1=I	51=LI	101=CI	151=CLI	201=CCI	251=CCLI	301=CCCI	351=CCCLI	401=CDI	451=CDLI
2=II	52=LII	102=CII	152=CLII	202=CCII	252=CCLII	302=CCCII	352=CCCLII	402=CDII	452=CDLII
3=III	53=LIII	103=CIII	153=CLIII	203=CCIII	253=CCLIII	303=CCCIII	353=CCCLIII	403=CDIII	453=CDLIII
4=IV=IIII	54=LIV	104=CIV	154=CLIV	204=CCIV	254=CCLIV	304=CCCIV	354=CCCLIV	404=CDIV	454=CDLIV
5=V	55=LV	105=CV	155=CLV	205=CCV	255=CCLV	305=CCCV	355=CCCLV	405=CDV	455=CDLV
6=VI	56=LVI	106=CVI	156=CLVI	206=CCVI	256=CCLVI	306=CCCVI	356=CCCLVI	406=CDVI	456=CDLVI
7=VII	57=LVII	107=CVII	157=CLVII	207=CCVII	257=CCLVII	307=CCCVII	357=CCCLVII	407=CDVII	457=CDLVII
8=VIII	58=LVIII	108=CVIII	158=CLVIII	208=CCVIII	258=CCLVIII	308=CCCVIII	358=CCCLVIII	408=CDVIII	458=CDLVIII
9=IX	59=LIX	109=CIX	159=CLIX	209=CCIX	259=CCLIX	309=CCCIX	359=CCCLIX	409=CDIX	459=CDLIX
10=X	60=LX	110=CX	160=CLX	210=CCX	260=CCLX	310=CCCX	360=CCCLX	410=CDX	460=CDLX
11=XI	61=LXI	111=CXI	161=CLXI	211=CCXI	261=CCLXI	311=CCCXI	361=CCCLXI	411=CDXI	461=CDLXI
12=XII	62=LXII	112=CXII	162=CLXII	212=CCXII	262=CCLXII	312=CCCXII	362=CCCLXII	412=CDXII	462=CDLXII
13=XIII	63=LXIII	113=CXIII	163=CLXIII	213=CCXIII	263=CCLXIII	313=CCCXIII	363=CCCLXIII	413=CDXIII	463=CDLXIII
14=XIV	64=LXIV	114=CXIV	164=CLXIV	214=CCXIV	264=CCLXIV	314=CCCXIV	364=CCCLXIV	414=CDXIV	464=CDLXIV
15=XV	65=LXV	115=CXV	165=CLXV	215=CCXV	265=CCLXV	315=CCCXV	365=CCCLXV	415=CDXV	465=CDLXV
16=XVI	66=LXVI	116=CXVI	166=CLXVI	216=CCXVI	266=CCLXVI	316=CCCXVI	366=CCCLXVI	416=CDXVI	466=CDLXVI
17=XVII	67=LXVII	117=CXVII	167=CLXVII	217=CCXVII	267=CCLXVII	317=CCCXVII	367=CCCLXVII	417=CDXVII	467=CDLXVII
18=XVIII	68=LXVIII	118=CXVIII	168=CLXVIII	218=CCXVIII	268=CCLXVIII	318=CCCXVIII	368=CCCLXVIII	418=CDXVIII	468=CDLXVIII
19=XIX	69=LXIX	119=CXIX	169=CLXIX	219=CCXIX	269=CCLXIX	319=CCCXIX	369=CCCLXIX	419=CDXIX	469=CDLXIX
20=XX	70=LXX	120=CXX	170=CLXX	220=CCXX	270=CCLXX	320=CCCXX	370=CCCLXX	420=CDXX	470=CDLXX
21=XXI	71=LXXI	121=CXXI	171=CLXXI	221=CCXXI	271=CCLXXI	321=CCCXXI	371=CCCLXXI	421=CDXXI	471=CDLXXI
22=XXII	72=LXXII	122=CXXII	172=CLXXII	222=CCXXII	272=CCLXXII	322=CCCXXII	372=CCCLXXII	422=CDXXII	472=CDLXXII
23=XXIII	73=LXXIII	123=CXXIII	173=CLXXIII	223=CCXXIII	273=CCLXXIII	323=CCCXXIII	373=CCCLXXIII	423=CDXXIII	473=CDLXXIII
24=XXIV	74=LXXIV	124=CXXIV	174=CLXXIV	224=CCXXIV	274=CCLXXIV	324=CCCXXIV	374=CCCLXXIV	424=CDXXIV	474=CDLXXIV
25=XXV	75=LXXV	125=CXXV	175=CLXXV	225=CCXXV	275=CCLXXV	325=CCCXXV	375=CCCLXXV	425=CDXXV	475=CDLXXV
26=XXVI	76=LXXVI	126=CXXVI	176=CLXXVI	226=CCXXVI	276=CCLXXVI	326=CCCXXVI	376=CCCLXXVI	426=CDXXVI	476=CDLXXVI
27=XXVII	77=LXXVII	127=CXXVII	177=CLXXVII	227=CCXXVII	277=CCLXXVII	327=CCCXXVII	377=CCCLXXVII	427=CDXXVII	477=CDLXXVII
28=XXVIII	78=LXXVIII	128=CXXVIII	178=CLXXVIII	228=CCXXVIII	278=CCLXXVIII	328=CCCXXVIII	378=CCCLXXVIII	428=CDXXVIII	478=CDLXXVIII
29=XXIX	79=LXXIX	129=CXXIX	179=CLXXIX	229=CCXXIX	279=CCLXXIX	329=CCCXXIX	379=CCCLXXIX	429=CDXXIX	479=CDLXXIX
30=XXX	80=LXXX	130=CXXX	180=CLXXX	230=CCXXX	280=CCLXXX	330=CCCXXX	380=CCCLXXX	430=CDXXX	480=CDLXXX
31=XXXI	81=LXXXI	131=CXXXI	181=CLXXXI	231=CCXXXI	281=CCLXXXI	331=CCCXXXI	381=CCCLXXXI	431=CDXXXI	481=CDLXXXI
32=XXXII	82=LXXXII	132=CXXXII	182=CLXXXII	232=CCXXXII	282=CCLXXXII	332=CCCXXXII	382=CCCLXXXII	432=CDXXXII	482=CDLXXXII
33=XXXIII	83=LXXXIII	133=CXXXIII	183=CLXXXIII	233=CCXXXIII	283=CCLXXXIII	333=CCCXXXIII	383=CCCLXXXIII	433=CDXXXIII	483=CDLXXXIII
34=XXXIV	84=LXXXIV	134=CXXXIV	184=CLXXXIV	234=CCXXXIV	284=CCLXXXIV	334=CCCXXXIV	384=CCCLXXXIV	434=CDXXXIV	484=CDLXXXIV
35=XXXV	85=LXXXV	135=CXXXV	185=CLXXXV	235=CCXXXV	285=CCLXXXV	335=CCCXXXV	385=CCCLXXXV	435=CDXXXV	485=CDLXXXV
36=XXXVI	86=LXXXVI	136=CXXXVI	186=CLXXXVI	236=CCXXXVI	286=CCLXXXVI	336=CCCXXXVI	386=CCCLXXXVI	436=CDXXXVI	486=CDLXXXVI
37=XXXVII	87=LXXXVII	137=CXXXVII	187=CLXXXVII	237=CCXXXVII	287=CCLXXXVII	337=CCCXXXVII	387=CCCLXXXVII	437=CDXXXVII	487=CDLXXXVII
38=XXXVIII	88=LXXXVIII	138=CXXXVIII	188=CLXXXVIII	238=CCXXXVIII	288=CCLXXXVIII	338=CCCXXXVIII	388=CCCLXXXVIII	438=CDXXXVIII	488=CDLXXXVIII
39=XXXIX	89=LXXXIX	139=CXXXIX	189=CLXXXIX	239=CCXXXIX	289=CCLXXXIX	339=CCCXXXIX	389=CCCLXXXIX	439=CDXXXIX	489=CDLXXXIX
40=XL	90=XC	140=CXL	190=CXC	240=CCXL	290=CCXC	340=CCCXL	390=CCCXC	440=CDXL	490=CDXC
41=XLI	91=XCI	141=CXLI	191=CXCI	241=CCXLI	291=CCXCI	341=CCCXLI	391=CCCXCI	441=CDXLI	491=CDXCI
42=XLII	92=XCII	142=CXLII	192=CXCII	242=CCXLII	292=CCXCII	342=CCCXLII	392=CCCXCII	442=CDXLII	492=CDXCII
43=XLIII	93=XCIII	143=CXLIII	193=CXCIII	243=CCXLIII	293=CCXCIII	343=CCCXLIII	393=CCCXCIII	443=CDXLIII	493=CDXCIII
44=XLIV	94=XCIV	144=CXLIV	194=CXCIV	244=CCXLIV	294=CCXCIV	344=CCCXLIV	394=CCCXCIV	444=CDXLIV	494=CDXCIV
45=XLV	95=XCV	145=CXLV	195=CXCV	245=CCXLV	295=CCXCV	345=CCCXLV	395=CCCXCV	445=CDXLV	495=CDXCV
46=XLVI	96=XCVI	146=CXLVI	196=CXCVI	246=CCXLVI	296=CCXCVI	346=CCCXLVI	396=CCCXCVI	446=CDXLVI	496=CDXCVI
47=XLVII	97=XCVII	147=CXLVII	197=CXCVII	247=CCXLVII	297=CCXCVII	347=CCCXLVII	397=CCCXCVII	447=CDXLVII	497=CDXCVII
48=XLVIII	98=XCVIII	148=CXLVIII	198=CXCVIII	248=CCXLVIII	298=CCXCVIII	348=CCCXLVIII	398=CCCXCVIII	448=CDXLVIII	498=CDXCVIII
49=XLIX	99=XCIX	149=CXLIX	199=CXCIX	249=CCXLIX	299=CCXCIX	349=CCCXLIX	399=CCCXCIX	449=CDXLIX	499=CDXCIX
50=L	100=C	150=CL	200=CC	250=CCL	300=CCC	350=CCCL	400=CD	450=CDL	500=D=Ↄ

www.tuomas.salste.net

Candidate Answer:

Language: JAVA
Total Execution Time: 0ms
Lines of code: 33

Not Passed, 0 marks
Compile time error

Code Submitted:

```
import java.io.*;
import java.util.*;

class Solution {
    public void solution(String S){
        //
       // Map for Roman values
    Map<Character , Interger> roman =new
    HashMap<>();
    roman.put('I', 1);
    roman.put('V', 5);
    roman.put('X', 10);
    roman.put('L', 50);
    roman.put('C', 100);
    roman.put('D', 500);
    roman.put('M', 1000);

    int n = s.length();
    int result = 0;

    for (int i = 0; i < n; i++) {
        int value = roman.get(s.charAt(i));
        if (i < n- 1 && value <
        roman.get(s.charAt(i + 1))) {
            result -= value;
        }else
    }

        //
    }
}

// Following is the part of the program and is provided as an assistance to read the input.
class Main {
    public static void main(String[] args) throws InterruptedException {
            Scanner sc = new Scanner(System.in);
            String S = sc.next();
            Solution s = new Solution();
            s.solution(S);
        }
    }
    // Map for Roman values
    Map<Character,
```

Question 4:

Total Time Spent Outside: **0 sec**
Total Move Count: **0**

Score: **0/20**
Time spent: **0 sec**

Numberland

In Numberland, the main occupation of its citizens is to perform tasks on numbers. One such important task is finding *interesting number sequences*.

As per the Numberland Research Institute, an *interesting sequence* is defined as a sequence of numbers that are consecutive, for instance: {10, 11, 12, 13} is an interesting sequence, and {2, 5, 8, 9} is not.

Mathematically, $\mathbf{n_{i+1} = n_i + 1}$

Now, citizens are given a set of distinct numbers to extract the length of the largest interesting sequence that can be obtained by rearranging numbers within the set.

Input:

```
Two lines.

* The first line contains N, the length of the input sequence.

* The second line of input consists of space separated N numbers, representing the set of numbers that m
```

Output:

```
* Single Integer, representing the largest sequence of consecutive numbers obtained after rearranging th
```

Constraints:

9 "d*N* "d1000

If K_i be the *i* th element in the sequence, 1 "dK_i "d2000

Candidate Answer:

Not Attempted

Conduct Metrics:

Criteria For Candidate Flagging:

Conduct Metrics shows the severity of window switch, window copy/paste, webcam face not detected, and webcam multiple face events

Candidate Report will be flagged in accordance with overall severity metrics

Sincere	Candidate's behavior (Code of Conduct) is authentic in order and does not impact the candidate's performance or validity of the test.
Moderately Suspicious	Candidate's behavior (Code of Conduct) has a low potential to affect the candidate's performance and validity of the test.
Extremely Suspicious	Candidate's behavior (Code of Conduct) critically impacts the candidate's performance and validity of the test.

Window Proctor History:

1. **Guidelines:**
   - Candidate Window Proctor History will capture the events of start/restart, resume, finish of test and events of tab/browser refreshing, closing, moving back to the previous screen
   - Time between each Refresh/Close/Back and Resume, also between Move-Out and Move-In will be considered as candidate time spent outside
2. **Event Time** will show the current event occurred time
3. **Event Time (from start)** will show the event occurred from start time of the test

Technical - 1

(Attempt - 1)

Total Time Spent :	Time Spent Outside Test Window :	Time Spent Within Test Window :	Total No Face Time :
14 min 9 sec	0 sec	14 min 9 sec	0 sec

▶	6:44:39 PM 28/08/2025 (a 0 sec from start)	Chrome	223.190.80.226	Windows
✓	6:58:49 PM 28/08/2025 (a 14 min 9 sec from start)	Chrome	223.190.80.226	Windows

Coding - 1

(Attempt - 1)

Total Time Spent :	Time Spent Outside Test Window :	Time Spent Within Test Window :	Total No Face Time :
30 min 14 sec	7 sec	30 min 6 sec	0 sec

▶	6:58:52 PM 28/08/2025 (a 0 sec from start)	Chrome	223.190.80.226	Windows
[←	7:04:58 PM 28/08/2025 (a 6 min 6 sec from start)	Chrome	223.190.80.226	Windows
→]	7:05:06 PM 28/08/2025 (a 6 min 14 sec from	Chrome	223.190.80.226	Windows

	start)			
	7:07:52 PM 28/08/2025 (a 9 min from start)	Chrome	223.190.80.226	Windows
	7:08:17 PM 28/08/2025 (a 9 min 24 sec from start)	Chrome	223.190.80.226	Windows
	7:29:06 PM 28/08/2025 (a 30 min 14 sec from start)			

IP Restricted Attempts:

Public IP	System IPs	Reason	Time of Restriction
No IP restricted attempt found.			

How to interpret the report?

The below color coding defines the performance of the candidate based on the percentage range in a section wise/overall:
Red: Low: The candidate has scored a percentage lesser than or equal to 40%
Blue: Moderate: The candidate has scored a percentage in the range of 40% to 70%
Green: High: The candidate has scored a percentage greater than 70%

Flags describes the cheating done by the candidate by moving away from the window while attempting this question.
Color of flag varies according to the level of cheating done by the candidate.

 Candidate has spent negligible amount of time outside of the test window.

 Candidate has spent less amount of time outside of the test window.

 Candidate has spent more time outside of the test window.

Speaking, Video Assessment And Profile Match Analysis

Confidence Analysis

(It depicts the level of confidence, fluency, and clarity the candidate possess)

- Candidate is highly confident (70% or above) in his/her ability to articulate his/her thoughts clearly when speaking.
- Candidate has moderate confidence (between 40% and 70%) in his/her ability to speak persuasively in a group setting.
- Candidate has low confidence (below 40%) in his/her ability to speak a language confidently.

Sentiment Analysis

(It depicts the nature of sentence)

- Positive statement
- Neutral statement
- Negative statement