Correct Marks : 3

Question Label : Short Answer Question

Using the Central Limit Theorem, find the approximate probability that in a random sample of 300 selected children at least 30 will have defective eye-sight. Enter the answer correct to 1 decimal place.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

0.5

CT

Section Id :	64065339710
Section Number :	4
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	14
Number of Questions to be attempted :	14
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065384337
Question Shuffling Allowed :	No

Is Section Default? :   null

Question Number : 44 Question Id : 640653586950 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "FOUNDATION LEVEL : COMPUTATIONAL THINKING (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE <u>TOP</u> FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531958496. ✔ YES

6406531958497. ✖ NO

Sub-Section Number :   2

Sub-Section Id :   64065384338

Question Shuffling Allowed :   No

Is Section Default? :   null

Question Number : 45 Question Id : 640653586951 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

Scores

RowNo	Name	Gender	DateOfBirth	CityTown	Mathematics	Physics	Chemistry	Total
0	Bhuvanesh	M	7 Nov	Erode	68	64	78	210
■ ■ ■								
29	Naveen	M	13 Oct	Vellore	72	66	81	219

Words

RowNo	Word	PartOfSpeech	LetterCount
0	It	Pronoun	2
■ ■ ■			
64	cane.	Noun	4

Library

RowNo	Name	Author	Genre	Language	Pages	Publisher	Year
0	Igniting Minds	Kalam	Nonfiction	English	178	Penguin	2002
■ ■ ■							
29	Malgudi Days	Narayan	Fiction	English	150	Indian Thought	1943

Olympics

Seq. No.	Name	Gender	Nationality	Host country	Year	Sport	Medal
0	Karnam Malleswari	F	Indian	Australia	2000	Weightlifting	Bronze
			- - -				
49	Michael Phelps	M	American	China	2008	Swimming	Gold

Three sample cards out of 30 for Shopping Bills dataset

Item List

SV Stores — Srivatsan ①

Item	Category	Qty	Price	Cost
Carrots	Vegetables/Food	1.5	50	75
Soap	Toiletries	4	32	128
Tomatoes	Vegetables/Food	2	40	80
Bananas	Vegetables/Food	8	8	64
Socks	Footwear/Apparel	3	56	168
Curd	Dairy/Food	0.5	32	16
Milk	Dairy/Food	1.5	24	36
				567

Sun General — Vignesh ⑭

Item	Category	Qty	Price	Cost
Phone Charger	Utilities	1	230	230
Razor Blades	Grooming	1	12	12
Razor	Grooming	1	45	45
Shaving Lotion	Grooming	0.8	180	144
Earphones	Electronics	1	210	210
Pencils	Stationery	3	5	15
				656

Big Bazaar — Sudeep ②

Item	Category	Qty	Price	Cost
Baked Beans	Canned/Food	1	125	125
Chicken Wings	Meat/Food	0.5	600	300
Cocoa powder	Canned/Food	1	160	160
Capsicum	Vegetables/Food	0.8	180	144
Tie	Apparel	2	390	780
Clips	Household	0.5	32	16
				1525

Options :

6406531958498. ✔ Useful Data has been mentioned above.

6406531958499. ✖ This data attachment is just for a reference & not for an evaluation.

Sub-Section Number : 3

Sub-Section Id : 64065384339

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 46 Question Id : 640653586952 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What will be the value of **mList** be at the end of the given pseudocode ?

```
1   L = [[1, 100, 'A'], [2, 99, 'B'], [3, 98, 'C'], [4, 97, 'D'], [5, 96, 'E']]
2   mList = [ ]
3   foreach element in L{
4       z = DoSomething(element)
5       mList = mList ++ [z]
6   }
7
8   Procedure DoSomething(X)
9       a = rest(X)
10      return(first(a))
11  End DoSomething
```

Options :

6406531958500. ✖
```
1   [[1, 100, 'A'], [2, 99, 'B'], [3, 98, 'C'], [4, 97, 'D'], [5, 96, 'E']]
```

6406531958501. ✖
```
1   [1, 100, 'A', 2, 99, 'B', 3, 98, 'C', 4, 97, 'D', 5, 96, 'E']
```

6406531958502. ✖
```
1   [ 'A',  'B', 'C',  'D',  'E' ]
```

6406531958503. ✔
```
1   [ 100, 99, 98, 97, 96 ]
```

Sub-Section Number :   4

Sub-Section Id :   64065384340

Question Shuffling Allowed :   Yes

Is Section Default? :   null

Question Number : 47 Question Id : 640653586953 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

The following pseudocode is executed using the "Words" dataset and **explode(X)** returns the list of letters in the word **X**. For example **explode**("sweet") will return ['s', 'w', 'e', 'e', 't']. What will **count** represent at the end of the execution?

```
1   count = 0, letterList = []
2   while(Table 1 has more rows){
3       Read the first row X in Table 1
4       letterList = explode(X.Word)
5       count = count + checkSomething(letterList)
6       Move X to Table 2
7   }
8
9   Procedure checkSomething(L)
10      lastLetter = first(L)
11      restList = rest(L)
12      foreach letter in restList{
13          if(letter == lastLetter){
14              return(1)
15          }
16          lastLetter = letter
17      }
18      return(0)
19  End checkSomething
```

Options :

6406531958504. ✖ Number of words with at most two consecutive occurrences of the same letter

6406531958505. ✔ Number of words with at least two consecutive occurrences of the same letter

6406531958506. ✖ Number of words with exactly two consecutive occurrences of the same letter

6406531958507. ✖ Number of words with no consecutive occurrences of the same letter

Question Number : 48 Question Id : 640653586960 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

The following pseudocode is executed using the "Words" dataset. What will **count** represent at the end of execution?

```
1   A = 10000, count = 0
2   L = []
3   while(Table 1 has more rows){
4       Read the first row X in Table 1
5       L = addSomething(L, X)
6       if(X.Word ends with a full stop){
7           if(length(L) == A){
8               count = count + 1
9           }
10          if(length(L) < A){
11              A = length(L)
12              count = 1
13          }
14          L = []
15      }
16      Move X to Table 2
17  }
18
19  Procedure addSomething(M, Y)
20      i = 1
21      while(i <= Y.LetterCount){
22          p = ith letter of Y.Word
23          if(not (member(M, p))){
24              M = M ++ [p]
25          }
26          i = i + 1
27      }
28      return(M)
29  End addSomething
```

Options :

6406531958537. � Number of sentences which have minimum number of letters

6406531958538. � Number of sentences which have minimum number of words

6406531958539. ✔ Number of sentences which have minimum number of distinct letters

6406531958540. � Number of sentences which have minimum number of distinct words

Sub-Section Number :	5
Sub-Section Id :	64065384341
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 49 Question Id : 640653586954 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Max. Selectable Options : 0

Question Label : Multiple Select Question

The following pseudocode is executed using the "Olympics" dataset. The template of "Olympics" dataset is attached in the beginnning of this paper. At the end of the execution, **medalDict** stores a dictionary with player's name as key mapped to another dictionary. The nested dictionary stores the medal type as key mapped to a list of years in which the player won that medal. For example if player Xyz has won a silver medal in 2006, a gold medal in 2008, and another silver medal in 2011, then

medalDict = {"Xyz" : {"Silver" : [2006, 2011], "Gold" : [2008]}, ... }

Assume that every player has a distinct name. But the pseudocode may have mistakes. Identify all such mistakes (if any). Assume that all statements not listed in the options below are free of errors. It is a Multiple Select Question (MSQ).

```
1   medalDict = {}
2   while(Table 1 has more rows){
3       Read the first row X in Table 1
4       if(isKey(medalDict, X.Name)){
5           if(not(iskey(medalDict[X.Name], X.Medal))){
6               medalDict[X.Name][X.Medal] = medalDict[X.Name][X.Medal] ++
    [X.Year]
7           }
8           else{
9               medalDict[X.Name][X.Medal] = [X.Year]
10          }
11      }
12      else{
13          medalDict[X.Name][X.Medal] = [X.Year]
14      }
15      Move X to Table 2
16  }
```

Options :

6406531958508. ✖ Line 1: Incorrect initialization of **medalDict**

6406531958509. ✔

Line 6: The current statement should be replaced by

```
1 | medalDict[X.Name][X.Medal] = [X.Year]
```

Line 9: The current statement should be replaced by

```
1 | medalDict[X.Name][X.Medal] = medalDict[X.Name][X.Medal] ++ [X.Year]
```

6406531958510. ✔

Line 13: The current statement should be replaced by

```
1 | medalDict[X.Name] = {X.Medal : [X.Year]}
```

6406531958511. ✔

6406531958512. ✘ No Mistakes

Question Number : 50 Question Id : 640653586959 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Let **LA** be a sorted list of integers in ascending order, and **X** be an integer. The procedure **insert(LA, X)** returns a list **LB** where X is added to **LA** such that the **LB** remains sorted. But the procedure may have mistakes. Identify all such mistakes (if any). Assume that all statements not listed in the options below are free of errors.

It is a Multiple Select Question (MSQ).

```
1   Procedure insert(LA, X)
2       LB = {}
3       flag = True
4       foreach A in LA{
5           if(flag){
6               if(X <= A){
7                   LB = LB ++ [A]
8                   flag = False
9               }
10          }
11          LB = LB ++ [A]
12      }
13      if(not flag){
14          LB = LB ++ [X]
15      }
16      return(LB)
17  End insert
```

Options :

6406531958531. ✔ Line 2: **LB** should be initiated as an empty list

6406531958532. ✖ Line 5: Conditional expression should use "not" operator

6406531958533. ✔ Line 7: **X** should be appended to the list **LB**

6406531958534. ✖ Line 11: **X** should be appended to the list **LB**

6406531958535. ✔ Line 13: Incorrect Conditional expression

6406531958536. ✖ Line 16: **LA** should be returned

Sub-Section Number :   6

Sub-Section Id :   64065384342

Question Shuffling Allowed :   Yes

Is Section Default? :   null

Question Number : 51 Question Id : 640653586955 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

The following pseudocode is executed using the "Olympics" dataset. The template of "Olympics" dataset is attached in the beginnning of this paper. What will **B** represent at the end of execution?

```
1   D = { }
2   while (Table 1 has more rows) {
3       Read the first row X in Table 1
4       D = updateDict(D, X.Sport)
5       Move X to Table 2
6   }
7   B = findAValue(D)
8
9   Procedure updateDict(D, a)
10      if(isKey(D, a)) {
11          D[a] = D[a] + 1
12      }
13      else {
14          D[a] = 1
15      }
16      return (D)
17  End updateDict
18
19  Procedure findAValue(D)
20      v = 0
21      foreach a in keys(D) {
22          if(D[a] > v) {
23              v = D[a]
24          }
25      }
26      return (v)
27  End findAValue
```

Options :

6406531958513. ✷ Number of players in each Sport

6406531958514. ✷ Total count of players in the Olympics dataset

6406531958515. ✔ Number of the players in the most frequent sport

6406531958516. ✷ Number of frequent sports

Sub-Section Number :   7

Question Number : 52 Question Id : 640653586956 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5

Question Label : Multiple Choice Question

The following pseudocode is executed on the "Words" dataset. What will **Count1** represent at the end of the execution?

```
1   D = {}
2   A = 0, Total = 0, Count = 0
3   while(Table 1 has more rows){
4       Read the first row X in Table 1
5       Total = Total + X.LetterCount
6       Count = Count + 1
7       if(isKey(D, X.Word)){
8           D[X.Word]["Freq"] = D[X.Word]["Freq"] + 1
9       }
10      else{
11          D[X.Word] = {}
12          D[X.Word]["Freq"] = 1
13          D[X.Word]["LC"] = X.LetterCount
14      }
15      if(D[X.Word]["Freq"] > A){
16          A = D[X.Word]["Freq"]
17      }
18      Move row X to Table 2
19  }
20  Avg = Total / Count
21  Count1 = 0, Count2 = 0
22  foreach k in Keys(D){
23      if(D[k]["Freq"] == A){
24          if(D[k]["LC"] > Avg){
25              Count1 = Count1 + 1
26          }
27          else{
28              Count2 = Count2 + 1
29          }
30      }
31  }
```

Options :

6406531958517. ✱ Number of maximum frequency words with letter count less than average letter count

6406531958518. ✔ Number of maximum frequency words with letter count greater than average letter count

6406531958519. ✱ Number of minimum frequency words with letter count less than average letter count

6406531958520. ✱ Number of minimum frequency words with letter count greater than average letter count

Sub-Section Number :	8
Sub-Section Id :	64065384344
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 53 Question Id : 640653586957 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

We have a non-empty list, **Location** that stores the city name in each card from the "Scores" dataset, sorted in alphabetical order. This results in many duplicates. The following procedure attempts to extract the unique list of cities, while preserving the sorted order. The pseudocode may have mistakes. Identify all such mistakes (if any). It is a Multiple Select Question.

```
1   uniqueList = []
2   uniqueList = uniqueList ++ [first(Location)]
3   prev = last(Location)
4   foreach x in rest(Location){
5       if(x != prev){
6           uniqueList = uniqueList ++ x
7       }
8       prev = x
9   }
```

Options :

6406531958521. ✔ Error in line 3

6406531958522. ✖ Error in line 4

6406531958523. ✖ Error in line 5

6406531958524. ✔ Error in line 6

6406531958525. ✖ Error in line 8

6406531958526. ✖ The pseudocode is error free

Question Number : 54 Question Id : 640653586958 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Alice and Bob are two friends. They decide to play a game of toss, in which each of them holds a coin. They toss their coins alternately for a fixed number of times, and the results of the series of tosses are recorded in **alist** (which is a list of lists). Each entry in the inner list results the outcome of one pair of coin tosses of the form **[i, j]** where **i** stores the outcome for Alice and **j** stores the outcome for Bob.

For example, **alist** = [['H', 'T'], ['T', 'T'], ['T', 'H']] implies that at the first trial Alice has a outcome 'H' and Bob has an outcome 'T', at the second trial Alice has a outcome 'T' and Bob has a outcome 'T', and at third trial Alice has a outcome 'T' and Bob has a outcome 'H'.

To determine the winner, a procedure **findWinner(x)** is called that accepts **aList** as a parameter and returns **c**. If **c** = 0, then Bob is the winner; if **c** = 1, then Alice is the winner; and **c** = 2 if it is a draw. Which of the following procedure(s) correctly identify/identifies the winner? It is a Multiple Select Question (MSQ).

Options :

6406531958527. ✖

```
1    Procedure findWinner(alist)
2        alice_sum = 0
3        bob_sum = 0
4        c = 2
5        foreach toss in alist{
6            if(first(toss)== 'H'){
7                alice_sum = alice_sum + 1
8            }
9            else{
10               bob_sum = bob_sum + 1
11           }
12       }
13       if(bob_sum < alice_sum){
14           c = 1
15       }
16       else{
17           c = 0
18       }
19       return(c)
20   end findWinner
```

```
1    Procedure findWinner(alist)
2        alice_sum = 0
3        bob_sum = 0
4        c = 2
5        foreach toss in alist{
6            if(first(toss)== 'H'){
7                alice_sum = alice_sum + 1
8            }
9            if(last(toss)== 'H'){
10               bob_sum = bob_sum + 1
11           }
12       }
13       if(bob_sum > alice_sum){
14           c = 0
15       }
16       if(bob_sum < alice_sum){
17           c = 1
18       }
19       return(c)
20   end findWinner
```

6406531958528. ✔

6406531958529. ✖

```
1    Procedure findWinner(alist)
2        alice_sum = 0
3        bob_sum = 0
4        c = -1
5        foreach toss in alist{
6            if(first(toss)== 'H'){
7                alice_sum = alice_sum + 1
8            }
9            if(last(toss)== 'H'){
10                bob_sum = bob_sum + 1
11            }
12        }
13        if(bob_sum > alice_sum){
14            c = 0
15        }
16        if(bob_sum < alice_sum){
17            c = 1
18        }
19        else{
20            c = 2
21        }
22        return(c)
23    end findWinner
```

```
1    Procedure findWinner(alist)
2        alice_sum = 0
3        bob_sum = 0
4        c = -1
5        foreach toss in alist{
6            if(first(toss)== 'H'){
7                alice_sum = alice_sum + 1
8            }
9            else{
10                bob_sum = bob_sum + 1
11            }
12        }
13        if(bob_sum < alice_sum){
14            c = 0
15        }
16        if(bob_sum > alice_sum){
17            c = 1
18        }
19        else{
20            c = 2
21        }
22        return(c)
23    end findWinner
```

Question Number : 55 Question Id : 640653586961 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider a dictionary **dict** = { 'A' : [1, 2, 3, 4, 5], 'B' : [6, 7, 8, 9], 'C' : 19 }. The following procedure **DoSomething(X)** is executed on the dictionary **dict**. What does the procedure **DoSomething(X)** return?

It is a Multiple Select Question.

```
1   procedure DoSomething(dict)
2     foreach i in keys(dict){
3       return(dict[i])
4     }
5   end DoSomething
```

Options :

6406531958541. ✔ The procedure may return [6,7,8,9]

6406531958542. ✖ The procedure always returns [1,2,3,4,5] since the key A comes first alphabetically

6406531958543. ✖ The procedure may return [1,2,3,5]

6406531958544. ✔ The procedure may return 19

Sub-Section Number :	9
Sub-Section Id :	64065384345
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 56 Question Id : 640653586962 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

The following pseudocode is executed using the "Library" dataset. The template of "Library" dataset is attached in the beginnning of this paper. **p** is a list of authors who have written books in English. **dict[X]** stores the number of books having at least 100 pages and written on or before 2000 by author **X**. Choose the correct code fragment(s) to complete the pseudocode.

It is a Multiple Select Question (MSQ).

```
1   dict = { }
2   foreach author in p{
3       dict[author] = 0
4   }
5   while(Table 1 has more rows){
6       Read the first row X from Table 1
7       ******************************
8       *         Fill the Code       *
9       ******************************
10      Move X to Table 2
11  }
```

Options :

```
1   if(isKey(dict, X.Author) and (X.Pages >= 100 and X.Year <= 2000)){
2       dict[author] = dict[author] + 1
3   }
```
6406531958545. ✔

```
1   if(isKey(dict, X.Author) or (X.Pages >= 100 and X.Year <= 2000)){
2       dict[author] = dict[author] + 1
3   }
```
6406531958546. ✖

```
1    C = False, D = False
2    if(isKey(dict, X.Author) and (X.Pages >= 100)){
3        C = True
4    }
5    if(isKey(dict, X.Author) and (X.Year <= 2000)){
6        D = True
7    }
8    if(C and D){
9        dict[author] = dict[author] + 1
10   }
```
6406531958547. ✔

6406531958548. ✖

```
1   C = False, D = False
2   if(isKey(dict, X.Author) and (X.Pages >= 100)){
3       C = True
4   }
5   if(isKey(dict, X.Author) and (X.Year <= 2000)){
6       D = True
7   }
8   if(C or D){
9       dict[author] = dict[author] + 1
10  }
```

Sub-Section Number :	10
Sub-Section Id :	64065384346
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653586963 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Question Numbers : (57 to 59)

Question Label : Comprehension

The following pseudocode is executed using the "Words" dataset. Answer the subquestions.

```
1   sList = [ ], wList = [ ], flag = 0, count = 0
2   while(Table 1 has more rows){
3       Read the first row X from Table 1
4       if(X.PartOfSpeech == "Adjective"){
5           count = count + 1
6       }
7       wList = wList ++ [X.Word]
8       if(X.Word ends with a full stop){
9           flag = count
10          sList = sList ++ [wList]
11          wList = [ ]
12          count = 0
13      }
14      Move row X to Table 2
15  }
```

Sub questions

Question Number : 57 Question Id : 640653586964 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

What will **flag** represent at the end of each sentence ?

Options :

6406531958549. ✲ Number of words in each sentence

6406531958550. ✔ Number of adjective(s) in each sentence

6406531958551. ✲ Number of non-adjective(s) in each sentence

6406531958552. ✲ Number of adjective(s) that are not repeated in each sentence

Question Number : 58 Question Id : 640653586965 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

wList will contain all words of last sentence of "Words" dataset at the end of execution of given pseudocode.

Options :

6406531958553. ✲ TRUE

6406531958554. ✔ FALSE

Question Number : 59 Question Id : 640653586966 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

The value of **length(sList)** will be same as the number of sentences in the "Words" dataset at the end of execution of given pseudocode.

Options :

6406531958555. ✔ TRUE

6406531958556. ✖ FALSE

DBMS

Section Id :	64065339711
Section Number :	5
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	16
Number of Questions to be attempted :	16
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065384347
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 60 Question Id : 640653586967 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0