Text Areas : PlainText

Possible Answers :

0.970 to 0.984

CT

Section Id :	64065328978
Section Number :	4
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	15
Number of Questions to be attempted :	15
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 :	64065363293
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 68 Question Id : 640653445507 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:SEMESTER 1/DIRECT ENTRY DIPLOMA : COMPUTATIONAL THINKING"

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 TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531484442. ✔ Yes

6406531484443. ✖ No

Question Number : 69 Question Id : 640653445508 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

SeqNo	Name	Gender	DateOfBirth	TownCity	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

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

Library

SeqNo	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

SeqNo	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

Options :

6406531484444. ✔ Useful Data has been mentioned above.

6406531484445. ✖ This data attachment is just for a reference & not for an evaluation.

Sub-Section Number : 2

Sub-Section Id : 64065363294

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 70 Question Id : 640653445509 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

What will be the value of **D** at the end of the execution of following pseudocode?

```
D = { 'a' : {'a': 5, 'b' : 4}, 'b': 1}
D['b'] = D['b'] + D['b']
```

Options :

6406531484446. ✖ D = { 'a' : {'a' : 5, 'b' : 4}, 'b' : 1}

6406531484447. ✔ D = { 'a' : {'a' : 5, 'b' : 4}, 'b' : 2}

6406531484448. ✖ D = { 'a' : {'a' : 10, 'b' : 8}, 'b' : 2}

6406531484449. ✖ D = { 'a' : {'a' : 5, 'b' : 8}, 'b' : 1}

Question Number : 71 Question Id : 640653445510 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

Let **dict** be a dictionary, then which of the following is not a valid value of **dict**?

Options :

6406531484450. ✖ **dict** = { 'a' : {'a' : 5, 'b' : 4}, 'b' : {'a' : 2}}

6406531484451. ✖ **dict** = { 'a' : {'a' : 'a', 'b' : 'b'}, 'b' : {'a' : 2}}

6406531484452. ✔ **dict** = { 'a' : {'a' : 5, 'a' : 4}, 'b' : {'a' : 2}}

6406531484453. ✖ **dict** = { 'a' : {'a' : 5, 'b' : 4}, 'b' : {'b' : 2}}

Sub-Section Number : 3

Sub-Section Id : 64065363295

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 72 Question Id : 640653445511 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

Let 'x', 'y', and 'z' be the only keys of dictionary **D** and **L** = **keys**(**D**). At the end of the execution of the following pseudocode, **flag** stores True.

```
flag = False
position = 0
foreach key in L{
    if((position == 1) and (key == 'y')){
        flag = True
    }
    position = position + 1
}
```

Choose the possible value of **L** from the given choices.

Options :

6406531484454. ✔ ['z', 'y', 'x']

6406531484455. ✖ ['y', 'x', 'z']

6406531484456. ✖ ['x', 'z','y']

6406531484457. ✖ ['z', 'x', 'y']

Sub-Section Number : 4

Sub-Section Id : 64065363296

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 73 Question Id : 640653445512 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

Let **timeList** be a list of pairs containing information about trains associated with a station **stn**. Specifically, each element in this list is a pair: [*Arrival*, *Departure*] (pair of arrival and departure time). If the arrival or departure time is empty, it is represented as "None". What does **count** represent at the end of the execution of the following pseudocode?

```
count = 0
foreach x in timeList{
    if(first(x) != "None" and last(x) != "None"){
        count = count + 1
    }
}
```

Options :

6406531484458. ✖ Number of trains for which **stn** is a starting station

6406531484459. ✖ Number of trains for which **stn** is an ending station

6406531484460. ✖ Number of trains for which **stn** is either a starting or an ending station

6406531484461. ✔ Number of trains for which **stn** is neither a starting nor an ending station

Question Number : 74 Question Id : 640653445513 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

Consider the following pseudocode, where **D** is a dictionary.

```
sum = 0
foreach key in keys(D){
    sum = sum + first(D[key])
}
```

Choose a statement regarding **D** from the given choices such that **sum** will always store a value greater than 0 at the end of the execution of the above code.

Options :

6406531484462. ✖ Each key of the dictionary **D** should be mapped to a positive integer.

6406531484463. ✖ Each key of the dictionary **D** should be mapped to a dictionary with each key mapped to a positive integer.

6406531484464. ✖ Each key of the dictionary **D** should be mapped to a non-empty list of integers.

6406531484465. ✔ Each key of the dictionary **D** should be mapped to a non-empty list of positive integers.

Question Number : 75 Question Id : 640653445514 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

Let **X** be a row from the "Words" table. Consider the following procedure.

```
Procedure isRich(X)
    vDict = {}
    i = 1, A = ''
    while(i <= X.LetterCount){
        A = ith letter in X.Word
        if(A is a vowel){
            vDict[A] = True
        }
        i = i + 1
    }
    if(length(keys(vDict)) >= 3){
        return(True)
    }
    return(False)
End isRich
```

The return value of **isRich**(Y) will be False if

Options :

6406531484466. ✔ **Y**.*Word* = "perseverance"

6406531484467. ✖ **Y**.*Word* = "computational"

6406531484468. ✖ **Y**.*Word* = "router"

6406531484469. ✖ **Y**.*Word* = "online"

Sub-Section Number :	5
Sub-Section Id :	64065363297
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 76 Question Id : 640653445515 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

Let **explode(W)** returns the list of letters in the word **W**. For example **explode**("common") will return ['c', 'o', 'm', 'm', 'o', 'n']. What will **count** store at the end of the execution of the following pseudocode?

```
count = 0, letterList = []
wordList = ["keep", "exploring", "and", "keep", "learning"]
foreach word in wordList{
    letterList = explode(word)
    lastLetter = '', flag = False
    foreach letter in letterList{
        if(letter is a vowel and letter == lastLetter){
            flag = True
        }
        lastLetter = letter
    }
    if(flag){
        count = count + 1
    }
}
```

Options :

6406531484470. ✖ 1

6406531484471. ✔ 2

6406531484472. ✖ 3

6406531484473. ✖ 4

Question Number : 77 Question Id : 640653445516 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. Assume that the rows in Table 1 are arranged in the increasing order of sequence numbers from top to bottom. At the end of the execution of the following pseudocode, **L** stores the list of nouns that appear immediately after an adjective. Choose the correct code fragment to complete the pseudocode.

```
L = []
A = "None"
Read the first row X in Table 1
A = X.PartOfSpeech
Move X to Table 2
while(Table 1 has more rows){
    Read the first row Y in Table 1
    **********
    **  Fill the code **
    **********
    A = Y.PartOfSpeech
    Move Y to Table 2
}
```

Options :

```
if(Y.PartOfSpeech == "Noun"){
    if(A == "Adjective"){
        L = L ++ [Y.Word]
    }
}
```

6406531484474. ✔

```
if((A == "Adjective") or (Y.PartOfSpeech == "Noun"){
        L = L ++ [Y.Word]
    }
}
```

6406531484475. ✖

```
if((A == "Noun") and (Y.PartOfSpeech == "Adjective"){
        L = L ++ [Y.Word]
    }
}
```

6406531484476. ✖

6406531484477. ✖

```
if(Y.PartOfSpeech == "Adjective"){
    if(A == "Noun"){
        L = L ++ [Y.Word]
    }
}
```

Sub-Section Number : 6

Sub-Section Id : 64065363298

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 78 Question Id : 640653445517 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Consider the procedure given below, where **aList** is a list of integers.

```
procedure cumulative(aList)
    sum = 0, cumuList = []
    foreach element in aList{
        sum = sum + element
        cumuList = cumuList ++ [sum]
    }
    return(cumuList)
end cumulative
```

At the end of the execution, which of the following option(s) would be correct? It is a Multiple Select Question (MSQ).

Options :

6406531484478. ✔ The first element of both the lists, **cumuList** and **aList**, will be same.

6406531484479. ✖ Number of elements in **cumuList** will be one lesser than that of **aList**

6406531484480. ✖ **cumuList** is a list of numbers in increasing order.

6406531484481. ✔ Number of elements in both lists, **cumuList** and **aList**, will be same

Sub-Section Number : 7

Sub-Section Id : 64065363299

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 79 Question Id : 640653445518 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4 Selectable Option : 0

Question Label : Multiple Select Question

Let **medalDict** be a dictionary with player's name as a key mapped to the list of medals associated with the player from the "Olympics" dataset. For example **medalDict** = {"xyz" : ["Silver", "Gold", "Gold"], ... }. In this example, the player xyz has won one Silver and two Gold medals.

At the end of the execution, **repeatMedals(medalDict)** returns the list of players who have won at least one type of medal more than one time. But the code 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 repeatMedals(medalDict)
2      repeatPlayers = []
3      foreach player in keys(medalDict){
4          tempDict = {}
5          foreach medal in medalDict[player]{
6              tempDict[medal] = True
7          }
8          if(length(keys(tempDict)) == length(medalDict[player])){
9              repeatPlayers = repeatPlayers ++ [player]
10         }
11     }
12     return(repeatPlayers)
13 End repeatMedals
```

Options :

6406531484482. ✖ Line 2: Incorrect initialization of **repeatPlayers**

6406531484483. ✖ Line 4: Incorrect initialization of **tempDict**

6406531484484. ✔ Line 8: Incorrect conditional statement to update **repeatPlayers**

6406531484485. ✖ No mistakes

Sub-Section Number : 8

Sub-Section Id : 64065363300

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 80 Question Id : 640653445519 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 5 Selectable Option : 0

Question Label : Multiple Select Question

The following pseudocode is executed using the "Olympics" dataset. 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(iskey(medalDict[X.Name], X.Medal)){
6              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 :

6406531484486. ✖ Line 1: Incorrect initialization of **medalDict**

6406531484487. ✔

Line 6: The current statement should be replaced by

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

Line 9: The current statement should be replaced by

6406531484488. ✖

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

Line 13: The current statement should be replaced by

6406531484489. ✔

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

6406531484490. ✖ No Mistakes

Sub-Section Number :	9
Sub-Section Id :	64065363301
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653445524 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 : (81 to 82)

Question Label : Comprehension

The following pseudocode is executed using the "Scores" dataset. At the end of the execution, **medalList** should store the list of sequence numbers of the students who have scored at least 200 total marks and have scored more than 80 marks at least in two subjects. Answer the given subquestions based on the pseudocode.

```
medalList = [], A = 0, sCount = False
while(Table 1 has more rows){
    Read the first row X in Table 1
    A = X.Total
    sCount = nSub(X.SeqNo)
    if((A >= 200) and sCount){
        medalList = medalList ++ [X.SeqNo]
    }
    Move X to Table 2
}
```

Sub questions

Question Number : 81 Question Id : 640653445525 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Selectable Option : 0

Question Label : Multiple Select Question

Which of the following statement(s) is(are) true about **nSub** based on the pseudocode mentioned in the main question ? It is a Multiple Select Question (MSQ).

Options :

6406531484494. ✔ **nSub** is a procedure which accepts the sequence number of a student and returns True if the student has scored more than 80 marks at least in two subjects otherwise returns False.

6406531484495. ✖ **nSub** is a procedure which accepts the sequence number of a student and returns the number of subjects in which the student has scored more than 80 marks.

6406531484496. ✖ **nSub** is a dictionary with sequence numbers of students mapped to the number of subjects in which the student has scored more than 80 marks.

6406531484497. ✖ **nSub** is a dictionary with sequence numbers of students mapped to True if the student has scored more than 80 marks at least in two subjects otherwise mapped to False.

Question Number : 82 Question Id : 640653445526 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

Let **M**, **P**, and **C** be the lists of sequence numbers of the students who have scored more than 80 marks in Mathematics, Physics, and Chemistry respectively. If **n** is the sequence number of a student then choose the correct implementation of **nSub**?

Options :

6406531484498. ✖

```
Procedure nSub(n)
    count = 0
    if(member(M, n)){
        count = count + 1
    }
    if(member(P, n)){
        count = count + 1
    }
    if(member(C, n)){
        count = count + 1
    }
    return(count)
End nSub
```

6406531484499. ✔

```
Procedure nSub(n)
    count = 0
    if(member(M, n)){
        count = count + 1
    }
    if(member(P, n)){
        count = count + 1
    }
    if(member(C, n)){
        count = count + 1
    }
    if(count >= 2){
        return(True)
    }
    return(False)
End nSub
```

```
nSub = {}
while(Table 1 has more rows){
    Read the first row X from Table 1
    count = 0
    if(member(M, X.SeqNo)){
        count = count + 1
    }
    if(member(P, X.SeqNo)){
        count = count + 1
    }
    if(member(C, X.SeqNo)){
        count = count + 1
    }
    if(count >= 2){
        nSub[SeqNo] = True
    }
    Move X to Table 2
}
```

6406531484500. ✖

```
nSub = {}
while(Table 1 has more rows){
    Read the first row X from Table 1
    count = 0
    if(member(M, X.SeqNo)){
        count = count + 1
    }
    if(member(P, X.SeqNo)){
        count = count + 1
    }
    if(member(C, X.SeqNo)){
        count = count + 1
    }
    nSub[X.SeqNo] = count
    Move X to Table 2
}
```

6406531484501. ✖

Sub-Section Number :	10
Sub-Section Id :	64065363302
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653445520 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 : (83 to 85)

Question Label : Comprehension

Let **Z** be a row in the "Words" table and **D** be a dictionary. Use the procedure given below for answering the given subquestions.

```
Procedure updateDict(Z, Dict)
    i = 1, x = ''
    while(i <= Z.LetterCount){
        x = ith letter of Z.Word
        if(not isKey(Dict, x)){
            Dict[x] = 1
        }
        else{
            Dict[x] = Dict[x] + 1
        }
        i = i + 1
    }
    return(Dict)
End updateDict
```

Sub questions

Question Number : 83 Question Id : 640653445521 Question Type : SA Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Short Answer Question

Let **X**.*Word* be "thinking". At the end of the execution of the following pseudocode, what will be the value of **length**(**keys**(**alphaDict**))?

```
alphaDict = {'t':2, 'c':1, 'e':1}
alphaDict = updateDict(X, alphaDict)
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

8

Question Number : 84 Question Id : 640653445522 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Short Answer Question

Let **X**.*Word* and **Y**.*Word* be "computational" and "thinking" respectively.The following pseudocode is executed using the "Words" dataset and the procedure **updateDict** mentioned in the main question.

```
firstDict = {}, secondDict = {}, commonDict = {}
firstDict = updateDict(X, commonDict)
secondDict = updateDict(Y, commonDict)
foreach key in keys(firstDict){
    if(isKey(secondDict, key)){
        if(firstDict[key] > secondDict[key]){
            commonDict[key] = firstDict[key]
        }
        else{
            commonDict[key] = secondDict[key]
        }
    }
}
```

At the end of the execution of above pseudocode, what would be the value of **length**(**keys**(**commonDict**))?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

3

Question Number : 85 Question Id : 640653445523 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Short Answer Question

Consider the dictionary **commonDict** created in the previous question. What would be the value of **commonDict**['i']?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

2

DBMS

Section Id :	64065328979
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 :	64065363303