question), Which point has the highest probability of being chosen as 3rd the cluster mean? Use the manhattan distance to compute the distances.

Options :

6406531562277. ✖ (3,1)

6406531562278. ✖ (4,7)

6406531562279. ✔ (-1,9)

6406531562280. ✖ (4,-2)

MLP

Section Id :	64065330338
Section Number :	12
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	24
Number of Questions to be attempted :	24
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 :	64065367711
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 193 Question Id : 640653470123 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 "DIPLOMA LEVEL : MACHINE LEARNING PRACTICE"

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 :

6406531562281. ✔ YES

6406531562282. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	64065367712
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 194 Question Id : 640653470124 Question Type : SA Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Short Answer Question

[String Matching] Enter the sequence of steps to be followed, in general, in end to end machine learning project. Enter the answer as a 6-character string. For example, entering the answer as (without quotes) 'BADCFE' implies that the first step is B, followed by the second step A and so on.

A. Select suitable model

B. Train the model

C. Pre-process the data

D. Collect data

E. Fine tune model

F. Present your solution

Response Type : Alphanumeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Answers Case Sensitive : No

Text Areas : PlainText

Possible Answers :

DCABEF

Sub-Section Number :	3
Sub-Section Id :	64065367713
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 195 Question Id : 640653470132 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

Which of the following APIs can be used to construct an ML pipeline for data preprocessing and modeling?

Options :

6406531562313. ✖ Pipeline

6406531562314. ✖ ColumnTransformer

6406531562315. ✖ FeatureUnion

6406531562316. ✔ All of these

Question Number : 196 Question Id : 640653470134 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

Consider following data:

```
data = [{'age': 4, 'height':96.0},
        {'age': 1, 'height':73.9},
        {'age': 3, 'height':88.9},
        {'age': 2, 'height':81.6}]
```

Which one of the following APIs can be used to extract features from the above data?

Options :

6406531562322. ✔ DictVectorizer

6406531562323. ✖ HashingVectorizer

6406531562324. ✖ FeatureHasher

Question Number : 197 Question Id : 640653470135 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

Consider following data:

$$\mathbf{X} = \begin{bmatrix} 8 & 3 \\ np.nan & 8 \\ 3 & np.nan \\ 10 & 6 \\ 7 & 7 \end{bmatrix}$$

What will be the output of the following code? (Assume necessary imports)

```
si = SimpleImputer(strategy='mean')
Xnew = si.fit_transform(X)
print(f'{Xnew[1,0]}, {Xnew[2,1]}')
```

Options :

6406531562325. ✔ 7,6

6406531562326. ✖ 6,6

6406531562327. ✖ 6,7

6406531562328. ✖ 7,7

Sub-Section Number : 4

Sub-Section Id : 64065367714

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 198 Question Id : 640653470126 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

Suppose that we load a data set that contains 1000 samples in a Pandas Dataframe. Each sample has 30 features. However, a few samples in the data set miss the values for all features. Therefore, those samples need to be dropped. Choose the method that removes such samples from the dataset?

Options :

6406531562288. ✖ drop(columns=['all'])

6406531562289. ✖ drop(how='all')

6406531562290. ✖ dropna()

6406531562291. ✔ dropna(how='all')

6406531562292. ✖ dropna.all()

Question Number : 199 Question Id : 640653470127 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 statement that each feature in the input data set has a (physical) meaning associated with it is

Options :

6406531562293. ✖ True, for all ML problems

6406531562294. ✔ True, for some ML problems

6406531562295. ✖ False, for all ML problems

6406531562296. ✖ can not be decided

Question Number : 200 Question Id : 640653470129 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

A company collects 40000 samples (examples) to build a Machine Learning model for an application. They decide to use 30% of the total samples for testing (to be stored in the variable *trainset*) and the rest 70% for training (to be stored in the variable *trainset*). They also want to sample the same set of samples across multiple runs. Which of the following line (statement) achieves this task? Assume that all samples are stored in the variable **data**.

Options :

6406531562301. ✖
```
testset, trainset = train_test_split(data, test_size=0.3,
                                     random_state=42)
```

6406531562302. ✖
```
trainset, testset = train_test_split(data, test_size=0.3)
```

6406531562303. ✔
```
trainset, testset = train_test_split(data, test_size=0.3,
                                     random_state=42)
```

6406531562304. ✖
```
testset, trainset = train_test_split(data, test_size=0.3)
```

Question Number : 201 Question Id : 640653470130 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

Which of the following utilities of sklearn.datasets helps us to get the realworld data from the internet?

Options :

6406531562305. ✖ load_

6406531562306. ✔ fetch_

6406531562307. ✖ generate_

6406531562308. ✖ get_

Question Number : 202 Question Id : 640653470133 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

Consider the following ML task/steps for a regression dataset:

1. Read the data from a file (named 'dataset.csv'). It has 7 columns. The last column is the target variable, the rest of them are numerical features.
2. Drop rows with missing values.
3. After removing samples with missing values split the data into training and test sets. Take about the first 80% of rows in the training set and the rest of them into the test set.
4. Train a simple linear regression model, with intercept, on the training set.
5. Report R2 score on the test set.

Which of the following code snippets correctly accomplishes the above task? Assume necessary imports.

Options :

6406531562317. ✔

```
data = pd.read_csv('dataset.csv')
data = data.dropna()
rows, cols = data.shape
data_train = data[:int(0.8*rows)]
data_test = data[int(0.8*rows):]
X_train = data_train[data.columns[:-1]]
y_train = data_train[data.columns[-1]]
X_test = data_test[data.columns[:-1]]
y_test = data_test[data.columns[-1]]

model = LinearRegression().fit(X_train,y_train)
model.score(X_test, y_test)
```

6406531562318. ✖

```
data = pd.read_csv('dataset.csv')
data = data.dropna()
rows, cols = data.shape
data_train = data[:int(0.2*rows)]
data_test = data[int(0.2*rows):]
X_train = data_train[data.columns[:-1]]
y_train = data_train[data.columns[-1]]
X_test = data_test[data.columns[:-1]]
y_test = data_test[data.columns[-1]]

model = LinearRegression().fit(X_train,y_train)
model.score(X_test, y_test)
```

6406531562319. ✖

```
data = pd.read_csv('dataset.csv')
data = data.dropna()
rows, cols = data.shape
data_train = data[:int(0.2*rows)]
data_test = data[int(0.2*rows):]
X_train = data_train[data.columns[:-1]]
y_train = data_train[data.columns[-1]]
X_test = data_test[data.columns[:-1]]
y_test = data_test[data.columns[-1]]

model = LinearRegression().fit(X_test,y_test)
model.score(X_test, y_test)
```

6406531562320. ✖

```
data = pd.read_csv('dataset.csv')
data = data.dropna()
rows, cols = data.shape
data_train = data[:int(0.8*rows)]
data_test = data[int(0.2*rows):]
X_train = data_train[data.columns[:-1]]
y_train = data_train[data.columns[-1]]
X_test = data_test[data.columns[:-1]]
y_test = data_test[data.columns[-1]]

model = LinearRegression().fit(X_train,y_test)
model.score(X_train, y_train)
```

6406531562321.

```
data = pd.read_csv('dataset.csv')
rows, cols = data.shape
data_train = data[:int(0.8*rows)]
data_test = data[int(0.2*rows):]
X_train = data_train[data.columns[:-1]]
y_train = data_train[data.columns[-1]]
X_test = data_test[data.columns[:-1]]
y_test = data_test[data.columns[-1]]

model = LinearRegression().fit(X_train,y_test)
model.score(X_train, y_train)
```

Question Number : 203 Question Id : 640653470140 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

Consider the following code where X_train, y_train is the training data. X_test, y_test is the test data.

```
from sklearn.linear_model import SGDRegressor
sgd= SGDRegressor(learning_rate='constant', eta0=1e-2)
sgd.fit(X_train, y_train)
sgd.predict(X_test)
score = sgd.score(X_test, y_test)
```

Which evaluation metric will be contained in the 'score'?

Options :

6406531562345. ✖ mean_squared_error

6406531562346. ✖ mean_absolute_error

6406531562347. ✔ R2_score

6406531562348. ✖ Accuracy

Question Number : 204 Question Id : 640653470142 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

How to make SGDRegressor stop after 1000 epochs?

Options :

6406531562353. ✖
```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(max_epoch=1000)
```

6406531562354. ✖
```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(stopping_criteria=1000)
```

6406531562355. ✔
```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(max_iter=1000)
```

6406531562356. ✖
```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(stop_after_iter=1000)
```

Question Number : 205 Question Id : 640653470143 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

Suppose we want to transform features in a dataset using polynomial transformation. The sklearn API provides the functionality in which of the following modules?

Options :

6406531562357. ✖ sklearn.dataset

6406531562358. ✖ sklearn.model_selection

6406531562359. ✔ sklearn.preprocessing

6406531562360. ✖ sklearn.featureSelection

6406531562361. ✖ sklearn.featureExtraction

Sub-Section Number : 5

Sub-Section Id : 64065367715

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 206 Question Id : 640653470136 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 following dataset:

HouseAge	AveRoom	Population	City	IncomeGroup
41.0	6.98	322.0	Delhi	Low
21.0	6.23	2401.0	Kolkata	High
52.0	8.28	496.0	Agra	Medium
52.0	NaN	558.0	Kolkata	Medium
52.0	6.28	565.0	Mumbai	Medium

Which one of the following code snippets will correctly preprocess above data? Assume necessary imports. The data is stored in a dataframe named X.

Options :

6406531562329. ✔

```
num_tranform = Pipeline(
    steps=[("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())]
)
cat_transfom = OneHotEncoder(handle_unknown="ignore")
ordinal_encoder = OrdinalEncoder()
preprocessor = ColumnTransformer(
    transformers=[
        ("num", num_tranform, ['HouseAge',
                               'AveRoom',
                               'Population']),
        ("cat", cat_transfom, ['City']),
        ("ord", ordinal_encoder, ['IncomeGroup'])
    ]
)

preprocessor.fit_transform(X)
```

6406531562330.

```
num_tranform = Pipeline(
    steps=[("scaler", StandardScaler())]
)
cat_transfom = OneHotEncoder(handle_unknown="ignore")
ordinal_encoder = OrdinalEncoder()
preprocessor = ColumnTransformer(
    transformers=[
        ("num", num_tranform, ['HouseAge',
                               'AveRoom',
                               'Population']),
        ("cat", cat_transfom, ['City']),
        ("ord", ordinal_encoder, ['IncomeGroup'])
    ]
)
preprocessor.fit_transform(X)
```

6406531562331.

```
num_tranform = Pipeline(
    steps=[("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())]
)
cat_transfom = OneHotEncoder(handle_unknown="ignore")
ordinal_encoder = OrdinalEncoder()
preprocessor = ColumnTransformer(
    transformers=[
        ("num", num_tranform, ['HouseAge',
                               'City',
                               'Population']),
        ("cat", cat_transfom, ['AveRoom']),
        ("ord", ordinal_encoder, ['IncomeGroup'])
    ]
)

preprocessor.fit_transform(X)
```

```
num_tranform = Pipeline(
    steps=[("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())]
)
cat_transfom = OneHotEncoder(handle_unknown="ignore")
ordinal_encoder = OrdinalEncoder()
preprocessor = ColumnTransformer(
    transformers=[
        ("num", num_tranform, ['IncomeGroup',
                               'AveRoom',
                               'Population']),
        ("cat", cat_transfom, ['City']),
        ("ord", ordinal_encoder, ['HouseAge'])
    ]
)

preprocessor.fit_transform(X)
```

6406531562332. ✖

Question Number : 207 Question Id : 640653470137 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

For a dataset with 1000 data points and 100 features, the following code will generate how many models during execution?

Note: X is the feature matrix and y is the target vector.

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import LeaveOneOut
from sklearn.linear_model import linear_regression
lin_reg = linear_regression()
loocv = LeaveOneOut()
score = cross_val_score(lin_reg, X, y, cv=loocv)
```

Options :

6406531562333. ✔ 1000

6406531562334. ✖ 100

6406531562335. ✖ 99

6406531562336. ✖ 999

Question Number : 208 Question Id : 640653470138 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 cross-validation strategy:

```
cv = ShuffleSplit(n_splits=40, test_size=0.3, random_state=0)
```

Assume we apply this strategy to some data set. Which of the following options is/are correct?

Options :

6406531562337. ✖ Every data point will be used only once for training.

6406531562338. ✖ Every data point will be used only once for validation.

6406531562339. ✖ The code will result in an error as $n\_splits * test\_size$ should be equal to 1.

6406531562340. ✔ None of these.

Question Number : 209 Question Id : 640653470139 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

Which of the following code is correct if we want the mean absolute error to be minimized during the computation of cross_val_score?

Options :

6406531562341. ✖
```
cross_val_score(lin_reg, X, y, cv=5,
                performance='neg_mean_absolute_error')
```

6406531562342. ✖
```
cross_val_score(lin_reg, X, y, cv=5,
                performance='mean_absolute_error')
```

6406531562343. ✔
```
cross_val_score(lin_reg, X, y, cv=5,
                scoring='neg_mean_absolute_error')
```

6406531562344. ✖
```
cross_val_score(lin_reg, X, y, cv=5,
                scoring='mean_absolute_error')
```

Question Number : 210 Question Id : 640653470144 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

We know that applying polynomial transformation to the samples is often helpful. Assume we imported all the required modules.

```
X = np.random.randn(2,2)
poly = PolynomialFeatures(degree=2,
                          interaction_only=True,
                          include_bias=False)
poly_X= poly.fit_transform(X)
```

What is the shape of the variable poly_X?

Options :

6406531562362. ✖ (2,2)

6406531562363. ✖ (2,6)

6406531562364. ✖ (6,2)

6406531562365. ✔ (2,3)

6406531562366. ✖ (3,2)

6406531562367. ✖ (2,5)

6406531562368. ✖ (5,2)

Question Number : 211 Question Id : 640653470145 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

A team has built a dataset for a regression problem. It contains 1000 samples. Each sample **x** is of size 2. All the features are binary, that is, $x_i \in \{0, 1\}$. The team decided to use polynomial feature transformation of degree 2 as follows,

```
X = np.random.randint(0,2,size=(10,2))
poly = PolynomialFeatures(degree=2)
poly_X= poly.fit_transform(X)
```

The transformed features are stored in the variable *poly_X*. There are N redundant column(s) (they are the exact copy of some column) in the variable *poly_X*. What is the value of N?

Options :

6406531562369. ✖ 1

6406531562370. ✔ 2

6406531562371. ✖ 3

6406531562372. ✖ 4

6406531562373. ✖ 0

Question Number : 212 Question Id : 640653470146 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 a regression problem with L2 regularization. Suppose we instantiate the model as shown below,

```
from sklearn.linear_model import Ridge
model = Ridge(alpha)
```

What is the range of 'alpha'?

Options :

6406531562374. ✖ $(-\infty, \infty)$

6406531562375. ✔ $(0, \infty)$

6406531562376. ✖ $(0, 1)$

6406531562377. ✖ No range,a constant 1.0

Sub-Section Number :	6
Sub-Section Id :	64065367716
Question Shuffling Allowed :	Yes

Is Section Default? : null

Question Number : 213 Question Id : 640653470125 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

(Multiple Select) Consider a data set shown below. The data set was loaded using Pandas and stored in the variable 'data' as a data frame.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

```
data_url = '''https://archive.ics.uci.edu/ml/machine-learning-databases/
wine-quality/winequality-red.csv'''
data = pd.read_csv(data_url, sep=";")
```

Suppose that we want to get the value of 'chlorides' of the third sample (2nd by index). Which of the following lines of code does this?

Options :

6406531562284. ✔ data.chlorides[2]

6406531562285. ✔ data['chlorides'][2]

6406531562286. ✖ data[2][4]

6406531562287. ✔ data.iloc[2,4]

Question Number : 214 Question Id : 640653470128 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

(MSQ) Suppose that we plot the histogram of numerical features in a data set. This reveals which of the following information?

Options :

6406531562297. ✔ Scale of the features

6406531562298. ✔ (left or right) Skew of the distribution

6406531562299. ✔ Modes in the distribution

6406531562300. ✔ Deduce total number of samples in the dataset

Question Number : 215 Question Id : 640653470131 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

Why is data preprocessing necessary?

Options :

6406531562309. ✔ While recording or noting down, the data collector forgot the values to be recorded and entered blanks.

6406531562310. ✖ Some columns have values only between 0 and 1.

6406531562311. ✔ The data is divided into multiple files and has to be combined.

6406531562312. ✖ The data has only numbers in all the columns.

Question Number : 216 Question Id : 640653470141 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Selectable Option : 0

Question Label : Multiple Select Question

[MSQ] Which of the following will produce constantly reducing learning rates?

Options :

6406531562349. ✖

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='constant',
                                eta0=1e-2)
```

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='invscaling',
                                eta0=1e-2,
                                power_t = 0.25)
```

6406531562350. ✔

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='adaptive',
                                eta0=1e-2)
```

6406531562351. ✖

```
from sklearn.linear_model import SGDRegressor
linear_regressor = SGDRegressor(learning_rate='optimal',
                                eta0=1e-2)
```

6406531562352. ✔

BDM

Section Id :	64065330339
Section Number :	13
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	8
Number of Questions to be attempted :	8
Section Marks :	15
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