Week-4, Graded Theory

Problem 1

Question

`A`, `B` and `C` are boolean variables. Which of the following boolean expressions are equivalent to `C`? This is a MSQ type question.

A	B	C
`False`	`False`	`False`
`False`	`True`	`True`
`True`	`False`	`False`
`True`	`True`	`True`

(a) `B`

(b) `(not(A) and B) or (A and B)`

(c) `(A and not(A)) or B`

(d) `((A or not(A)) and B)`

(e) None of these

Answer

(a), (b), (c), (d)

Solution

All boolean expressions are equivalent. A user can follow any of the expressions and achieve the same result. There are various ways to transform expressions in one form to another, for example Sum of Products (SOP), Products of Sum (POS) etc. However, this simple exercise does not require these structured approach. It can be figured out by carefully observing the expressions.

The below code shows all the expressions results in the same output:

```
X = [[False, False, False],[False, True, True],[True, False, False],[True, True, True]]
for A, B, C in X:
    print(B, (not(A) and B) or (A and B), ((A and not(A)) or B), ((A or not(A)) and B))
```

Problem 2

Question

Match the statements with respective output in the below table.

`string = "this is old python course"`

Statement	OUTPUT
(a) `string.count("is")`	(1) `True`
(b) `string.replace("old", "new")`	(2) `12`
(c) `string.index("python")`	(3) `'this is new python course'`
(d) `string.startswith("this")`	(4) `2`
(e) `string.isalnum()`	(5) `False`
(f) `string.title()`	(6) `'This Is Old Python Course'`
(g) `string.strip("this is old")`	(7) `'python course'`

Select the correct option. This is a MCQ type question.

(A) (a) - (4), (b) - (3), (c) - (2), (d) - (1), (e) - (5), (f) - (6), (g) - (7)

(B) (a) - (5), (b) - (3), (c) - (2), (d) - (1), (e) - (4), (f) - (7), (g) - (6)

(C) (a) - (1), (b) - (3), (c) - (2), (d) - (5), (e) - (4), (f) - (6), (g) - (7)

(D) (a) - (4), (b) - (2), (c) - (3), (d) - (1), (e) - (6), (f) - (5), (g) - (7)

Answer

(A)

Solution

(a) `string.count("is")` counts the number of times substring `"is"` appears in `"this is old python course"`.

(b) `string.replace("old", "new")` replaces `"old"` to `"new"` in `"this is old python course"`.

(d) `string.startswith("this")` returns bool literal `True`, if the string starts with `"this"`.

(e) `string.isalnum()` returns bool literal `True`, if the string contains only alphabets and numbers.

(f) `string.title()` returns a string where the first character of each word is capitalized.

(g) `string.strip("this is old")` removes the string value passed to it from beginning and end of the original string `"this is old python course"`.

Problem 3

Question

What should be the value of `a`, `b`,`c`,`d` to print the list of first 10 positive even numbers regardless of the order. This is a MSQ type question.

```
for num in range(a, b, c):
    num = num + d
    print(num)
```

(a) `2, 21, 2, 0`

(b) `20, 0, -2, 1`

(c) `0, 19, 2, 2`

(d) `22, 2, -2, -2`

(e) None of these

Answer

(a), (c), (d)

Solution

The `range(a, b, c)` gives an arithmetic sequence given below which starts at `a`, elements are incremented by value `c` till it reaches `b`. The endpoint `b` is not included.

```
a, a + c, a + 2 * c, a + (n - 1) * c; where n is an arbitrary positive
integer, the last value in this sequence should be less than `b`.
```

For a = 0, b = 19, c = 2, the resulting sequence is $0, 2, 4, 8, 10, 12, 14, 16, 18$.

In for loop, at each iteration, the variable `num` is assigned the value from this sequence. Inside the body of the loop, `num` is incremented by `2` and printed.

In the very first iteration, `num` is set to `0`. The value is then incremented by `2` inside the loop and the same is printed. On the next iteration, the loop variable `num` stores the second element from the sequence which is `2`, this is incremented by `2` inside the body and printed on the console. This repeats for the rest of the iterations.

```
a, b, c, d = 0, 19, 2, 2
for num in range(a, b, c):
    print("Loop variable:", num, end = ", ")
    num = num + d
    print("Updated value: ", num)
```

Problem 4

Question

What should be the value of `x`, if the following code prints `python` on the console. This is a MSQ type question.

```
for num in x:
    print(num, end="")
```

(a) `["python",]`

(b) `["p", "y", "t", "h", "o", "n"]`

(c) `"python"`

(e) None of these

Answer

(a), (b), (c)

Solution

Option (a) prints the first element of the list, which is the string `python`. In option (b), all the `6` elements are printed one after another, therefore the text 'python' displayed on the console. In option (c), the characters are printed from the string `python` one after another. The string `python` behaves in the similar way as the list `["p", "y", "t", "h", "o", "n"]` here.

Problem 5

Question

A data structure is called heterogeneous if it can hold elements of different data types. Otherwise it is called homogeneous. Does a list need to be homogeneous?

(a) Yes, it needs to be homogeneous

(b) No, it need not be homogeneous

Answer

(b)

Solution

The `list` allows different data types as its element. Hence, it is a heterogeneous data type.

Problem 6

Question

Which of the following options are correct? This is a MSQ type question.

```
fruit_list = ["apple", "banana", "orange", "apple", "pineapple" ]
```

(a) `fruit_list.remove("apple")` removes the first occurrence (from the left) of the string "apple"

(b) `fruit_list.remove("apple")` removes the last occurrence (from the left) of the string "apple"

(c) The length of `fruit_list` becomes 4 after the execution of `fruit_list.remove("apple")`

(d) The datatype of `fruit_list` is `int`

(e) `fruit_list.append("apple")` removes the last occurrence (from the left) of the string "apple"

Answer

(a), (c)

Solution

`fruit_list.remove("apple")` removes the last occurrence of the string "apple". There were elements in the list. The length of the list reduced to 4 after removing one element.

Problem 7

Question

You are given two lists:

```
list_1 = ["I play", "You play"]
list_2 = ["Badminton", "Cricket"]
```

Which of the following option(s) gives the below output. This is a MSQ type question.

```
I play Badminton
```

(a) `print(list_1[0] + " " + list_2[0])`

(b) `print("{a} {b}".format(a = list_1[0], b = list_2[0]))`

(c)

```
for x in list_1[:-1]:
    for y in list_2[:-1]:
        print(x, y)
```

(d) None of these

Answer

(a), (b), (c)

Solution

All the above options are correct. The first element of the two lists `list_1` and `list_2` are accessed and `'I play Badminton'` is printed on the console.

Common data for problems 8 and 9

Assume that `L` is a non-empty list of positive integers. Also assume that the list is a distinct collection of numbers, i.e., no two numbers are alike. Consider the following code.

```
S = 0
for x in L:
    S += x

flag = False
y = -1
for x in L:
    if x * len(L) == S:
        flag = True
        y = x
        break
```

Problem 8

Question

If `flag` is `True` at the end of execution of the code given above, which of the following statements are true? Note that the options should be true for any list `L` that satisfies the conditions given in the common data. Multiple options could be correct.

(a) `y` is an element in the list `L`

(b) `y` is the smallest number in the list

(c) `y` is the greatest number in the list

(d) `y` is the average (arithmetic mean) of the numbers in the list

(e) `y` is the element at index `len(L) // 2` in the list `L`

Answer

(a), (d)

Solution

The main idea behind the code is as follows. `S` holds the sum of the elements in the list `L`. Line-8 inside the `for` loop essentially amounts to this:

> If each element in the list is replaced by `x`, will the sum of the elements still be equal to `S`?

If the answer to this question is yes, then the `flag` is changed to `True` and `y` is set to `x` and we break out of the loop. If the answer to this question is `False`, then we simply move onto the next iteration. At the end of execution, if `flag` remains `False`, then no such element `x` with the required characteristic is found.

Moving on, if `x * len(L) == S`, for some element `x` in the list `L`, then `y` is this element at the end of execution of the above code. Then we see that `y` is nothing but `S // len(L)`. We can use the floor division operator because `S` is divisible by `len(L)`. So, `y` is the average of the numbers in the list.

Problem 9

Question

Assume that `L` is a list of the first n positive integers, where $n > 0$. Under what conditions will the variable `flag` be `True` at the end of execution of the code given above?

(a) `n` is an odd integer

(b) `n` is an even integer

Answer

(a)

If `L` is a list of the first n positive integers then, $S = \frac{n(n+1)}{2}$. For `flag` to be true, there must be some number x between 1 and n that satisfies the following equation:

$$x \cdot n = \frac{n(n+1)}{2}$$

We immediately see that $x = \frac{n+1}{2}$. Since the list contains only integers, n would have to be odd.

Problem 10

Question

Let `X` be a vector. We wish to compute the dot product of `X` with itself and store it in a variable called `dp`. Select the correct code snippet(s) from the options given below. It is a Multiple Select Question (MSQ).

(a)

```
dp = 0
# Dot product of a vector with itself
# is just the sum of squares of the
# components of the vector
for elem in X:
    dp += elem * elem
```

(b)

```
dp = 0
# this is also a valid method
for i in range(len(X)):
    dp += X[i] ** 2
```

(c)

```
dp = 0
# this is WRONG
# range(len(X)) is the right command
for i in range(X):
    dp += X[i] * X[i]
```

(d)

```
i, dp = 0, 0
# this is WRONG
# it should be i < len(X)
while i <= len(X):
    dp += X[i] ** 2
    i += 1
```

(e)

```
i, dp = 0, 0
#
while i < len(X):
    dp += X[i] ** 2
    i += 1
```

Answer

(a), (b), (e)

Solution

Check the comments in the code.