

Q1)

Correct answer(s):

- 5

Explanation:

- This question focusses on producing the required output using one of the provided while conditions
- If you look at the given output, you know the loop should run its content 4 times to produce the output
- That means the while condition is evaluated 5 times, producing True, True, True, True and False when the loop is run
- Here are the values of each condition when the loop is run:
 - `response == "Y"`
True, True, True, True and False
 - `response != "Y"`
False, False, False, False and True
 - `response == "N"`
False, False, False, False and True
 - `response != "N"`
True, True, True, True and False
 - `number_of_candy_bars != 4`
True, True, True, True and False
 - `number_of_candy_bars == 4`
False, False, False, False and True
 - `number_of_candy_bars < 4`
True, True, True, True and False
 - `total_cost != 28`
True, True, True, True and False
 - `total_cost < 35`
True, True, True, True and True
 - `total_cost >= 28`
False, False, False, False and True
- As you can see, 5 of the conditions allow the loop to run its content four times

Q2)

Correct answer(s):

- First blank:
`balls.append(number)`
- Second blank:
`index = random.randint(0, len(balls) - 1)`
- Third blank:
`balls.remove(number)`

Explanation:

- For the first blank:
 - The objective is to add the numbers, from 1 to 49, to the `balls` list
 - One way is to simply add each number to the end of the list and therefore the `.append()` method can be used
 - Alternatively, you can also use the `.insert()` method to add each number, e.g. using `balls.insert(0, number)`
- For the second blank:
 - The objective is to randomly pick a number from the `balls` list
 - The `index` variable should contain the item index of the randomly picked number
 - Since the indices of the items in the list start from 0 to `len(balls)-1`, you should use `len(balls)-1` in the blank so that `random.randint()` covers the entire range of the indices
- For the third blank:
 - The objective is to remove the number that gets randomly picked so that the number will not be repeated later
 - That can be achieved by removing the number from the `balls` list and hence the use of the `.remove()` method

Q3)

Correct answer(s):

- 1:
- 1::
- 1::1
- 1:len(numbers)
- 1:len(numbers):1

Explanation:

- By examining the loop condition:

```
while len(numbers) > 0:
```

you can see the loop runs when there are items inside the `numbers` list
- In other words, the loop stops when the list is empty
- Therefore, the loop content needs to remove some items from `numbers` so that the loop condition can work correctly
- The loop does that using the last line of code inside the loop, which is the one that you need to work on:

```
numbers = numbers[ SEE_QUESTION ]
```
- To remove items from the list, one way to do that is by using slicing, i.e.:

```
numbers = numbers[ <start> : <end> : <step> ]
```
- From the if statement at the start of the loop, the number that the user is interested in, i.e. the `number` variable, is always compared against the first item of the list
- To make the program work,
 - you need to make sure the number is compared against every item in the list, not only the first number
 - that means the first item of the list cannot remain the same throughout the loop
- You can achieve this by removing the first item from the `numbers` list
- The simplest way to do that is by writing this slice notation:

```
numbers = numbers[ 1 : ]
```
- You do not need to provide the end number and the step number because by default, slicing will extract items until the end of the list with a step number of 1
- You can also use the alternative answers as shown above in the correct answers