Fill out the missing code

Q1 The following code should keep asking the user to enter a number **until the user enters the number 5**. Write the missing condition of the while loop.

```
number = int(input('Enter a number>'))
while |__Q1__| :
     number = int(input('Enter a number>'))
```

```
a.   number == 5
b.   number <= 5
c.   number != 5
d.   number >= 5
e.   number // 5
```

Q2 The following code should keep asking the user to enter a number until the **user enters a number other than** 5. Write the missing condition of the while loop.

```
number = int(input('Enter a number>'))
while |__Q2__|:
     number = int(input('Enter a number>'))
```

```
a.   number == 5
b.   number <= 5
c.   number != 5
d.   number >= 5
e.   number // 5
```

Q3 The following code should keep asking the user to enter a number untils the **user enters a number that is either less than 1 or greater than 5**. Write the missing condition of the while loop.

```
number = int(input('Enter a number>'))
while |__Q3__|:
     number = int(input('Enter a number>'))
```

```
a.   number <=1 and number >=5
b.   number >=1 and number <= 5
c.   number >=1 or number <= 5
d.   number != 5
e.   number == 6
```

Q4 The following code prints all numbers starting from 1 and going up to 12. Write the missing part of the for loop.

```
for number in |__Q4__| :
    print(number)
```

```
a. range(1,12)
b. len(12)
c. range(13)
d. [1, 2, 3, 4, 5, 6] * 2
e. [1, 2, 3, 4, 5, 6] + [7, 8, 9, 10, 11, 12]
```

Q5 The following code prints all odd numbers between 1 and 9 (both endpoints inclusive). Write the missing part of the if statement.

```
for number in range(1,10):
    if |__Q5__| :
        print(number)
```

```
a.   number % 2 < 0
b.   number % 2 <= 0
c.   number % 2 != 0
d.   number % 2 == 0
```

Q6 What is the output for the following program segment:

```
x = 2
y = 6
if x!=0:
    if y == 2:
        print(x)
    elif y % 2 == 1:
        print(y*2)
    elif y //2 == 3:
        print(y//2)
    else:
        print(x)
```

a. 3
b. 2
c. 12
d. 4
e. 5

Q7 What is the output for the following program segment?

```
numbers = [ ]
x = 1
while x != 0 :
   x = -x
   numbers.append(x)
   x = x - x
print(numbers)
```

a. [-1, -2]
b. [-1]
c. [-1, 0]
d. [0]
e. [0, - 1, -2]

Q8 What is the output?

```
strings = [ ['ab'] ]
print(len(strings[0]))
```

a. 0
b. 2
c. 1
d. 3
e. 4

Q9 What is the output?

```
alist=[[0,1],['Fred','Barney']]
blist = alist[alist[0][1]]
print(blist[1][0])
```

a. 1

b. Fred

c. Barney

d. B

e. F

Q10 What is the output?

```
grid = [ ["a", "b", "c"], ["d", "e", "f"], ["g", "h", "i"] ]
for i in range(len(grid)):
    if i % 2 == 1:
        print(grid[i][i])
```

a. a
b. b
c. c
d. d
e. e
f. f
g. g
h. h
i. i

Q11 What is the output?

```
string = ''
for x in ['ab','cd']:
    for y in x:
        string = string + y
print(string, end ='')
```

a. abcd
b. ab
c. cd
d. abcdabcd
e. abcddcba

Q12 What is the output?

```
def foo(alist):
    alist.append(4)
def main():
    blist = [2, 6, 8]
    alist = foo(blist)
    print(alist)
main()
```

a. [2, 6, 8]

b. [2, 6, 8, 4]

c. [4]

d. [4, 2, 6, 8]

e. None

Q13 What is the output?

```
def change(alist):
    blist = alist[2]
    blist.append(5)
def main():
    mylist = [1,2,[3]]
    change(mylist)
    print(mylist)
main()
```

a. [1,2,[3]]

b. [2, 5]

c. [1,2,[3,5]]

d. [3, 5]

e. [1,2,[3],5]

Q14 What is the output?

```
def change(greeting):
    greeting.upper()
    return greeting
def main():
    result = change('welcome')
    print(result)
main()
```

```
a. Welcome
b. WELCOME
c. None
d. welcome
e. greeting
```

Q15 What is the output?

```
def main():
    a_list = [0, 5, 10]
    result = test_function(a_list, 5)
    print(a_list)

def test_function(a_list, increment):
    a_list.reverse()
    for i in range(0, len(a_list)):
        a_list[i] = a_list[i] + increment
main()
```

```
a. [0, 5, 10]
b. [15, 10, 5]
c. [0, 25, 50]
d. [5, 10, 15]
e. None
```

Q16 What is the output?

```
def main():
    a_list = [0,5,10]
    result = test_function(a_list, 5)
    print(result)

def test_function(a_list, increment):
    a_list.reverse()
    for a in a_list:
        a = a + increment
main()
```

a. [0, 5, 10]

b. [15, 10, 5]

c. [0, 25, 50]

d. [5, 10, 15]

e. None

Q17 What is the output?

```
def main():
    a_list = [0,5,10]
    result = test_function(a_list, 5)
    print(~~alist~~) a_list

def test_function(a_list, increment):
    b_list = []
    for i in range(0, len(a_list)):
        b_list.append(a_list[i] + increment)
    a_list = b_list
main()
```

a. [0, 5, 10]

b. [15, 10, 5]

c. [0, 25, 50]

d. [5, 10, 15]

e. None

Q18 What is the output?

```
d1 = {'red' : 3, 'blue' : 5, 'white' : 6}
print(d1.get('green',0),end = ' ')
print(d1.get(0,'red'),end = ' ')
print(d1.get('blue'),end = ' ')
print(d1.get('white and red'),end = ' ')
```

```
a. 0 red 5 None
b. 0 3 5 pink
c. 0 red 5 63
d. green 3 5 None
e. green red blue white red
```

Q19 What is the output?

```
d1 = {'red' : 3, 'blue' : 5, 'white' : 6}
print(d1.pop('green',0),end = ' ')
print(d1.pop(0,'red'),end = ' ')
print(d1.pop('blue'),end = ' ')
print(d1.pop('white and red',None),end = ' ')
print(d1)
```

```
a. 0 red 5 None {'red': 3, 'white': 6}
b. 0 3 5 pink {'red' : 3, 'blue' : 5, 'white' : 6}
c. 0 red 5 63 {'red' : 3, 'blue' : 5}
d. green 3 5 None {}
e. green red blue white red {'red' : 3, 'blue' : 5, 'white' : 6}
```

Q20 What is the output?

```
def a():
    print('A',end = ' ')
    return True
def b():
    print('B',end = ' ')
    return True

print(a() or b())
```

```
a. A True
b. A True B True
c. A True B
d. A B
e. True True
```

Q21 A circle of radius *r* needs to be drawn at a random location inside a *w x h* window such that it is completely inside the window. Which one of the following expressions would you choose for the x and y coordinates of the center of the circle:

```
a.(random.randint(r, w - r), random.randint(r, h- r))
b. (random.randint(0, w - r), random.randint(0, h- r))
c.(random.randint(r, w), random.randint(r, h))
d.(random.randint(0, w), random.randint(0, h))
```

Q22 Refer to the Poke The Dot implementation. While a game is running, if a player clicks inside a window, the dot moves to a random location inside the window. Which one of the following conditional expressions would you choose for the `handle_events` method of Game class.

```
a. if event.type == pygame.MOUSEBUTTONUP and self.continue_game = True:
b. if event.type == pygame.MOUSEBUTTONUP and self.continue_game == True:
c. if event.type == pygame.MOUSEMOTION and self.continue_game:
d. if event.type == pygame.MOUSEBUTTONUP and self.close_clicked:
e. if event.type == pygame.MOUSEBUTTONUP and continue_game = True:
```

Q23 Which one of the following expressions checks if a dot of radius *r* and center *c* has moved past the top edge of a *w x h* window ?

```
a. c < r
b. c[0] < r
c. c[1] < r
d. c[0] + r > w
e. c[1] + r > h
f. c[0] > r
g. c[1] > r
```

Q24 Which one of the following expressions checks if a dot of radius *r* and center *c* has moved past the bottom edge of a *w x h* window ?

```
a. c < r
b. c[0] < r
c. c[1] < r
d. c[0] + r > w
e. c[1] + r > h
f. c[0] > r
g. c[1] > r
```

Q25 Assume there exists a matrix with m rows and n columns, where m and n are both greater than or equal to 2. What does the following code print?

```
for a in range(0,m):
    for b in range(0,n):
```

```
    if a == 0 and b == n-1:
        print(matrix[a][b])
```

a. number in the top right corner of the the matrix
b. number in the bottom right corner of the the matrix
c. number in the top left corner of the the matrix
d. number in the bottom left corner of the the matrix