Question 2

John is working on automating the rental process of a convention hall booking. He has identified two modules: interior decoration module and exterior decoration module. In both the modules, John w
include the logic for validating the number of days the convention hall is getting hired for.
Given below are the three possible solutions to John's requirement

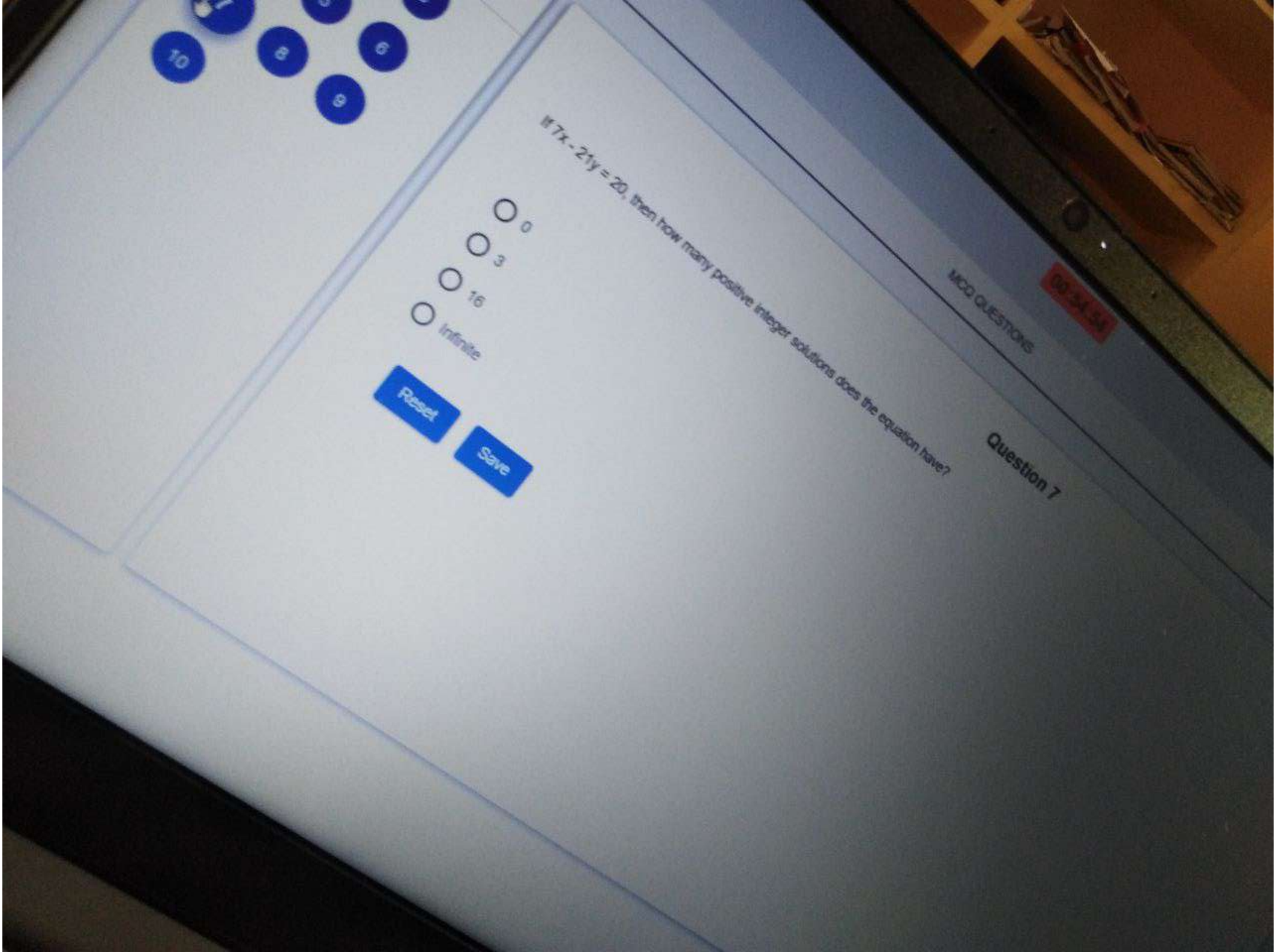
Solution-1. Create one function validate_number_of_days() and reuse it on requirement basis
Solution-2. Write the logic for validating number of days in both the modules along with business logic
Solution-3. Create two different validation functions with the same logic in each module

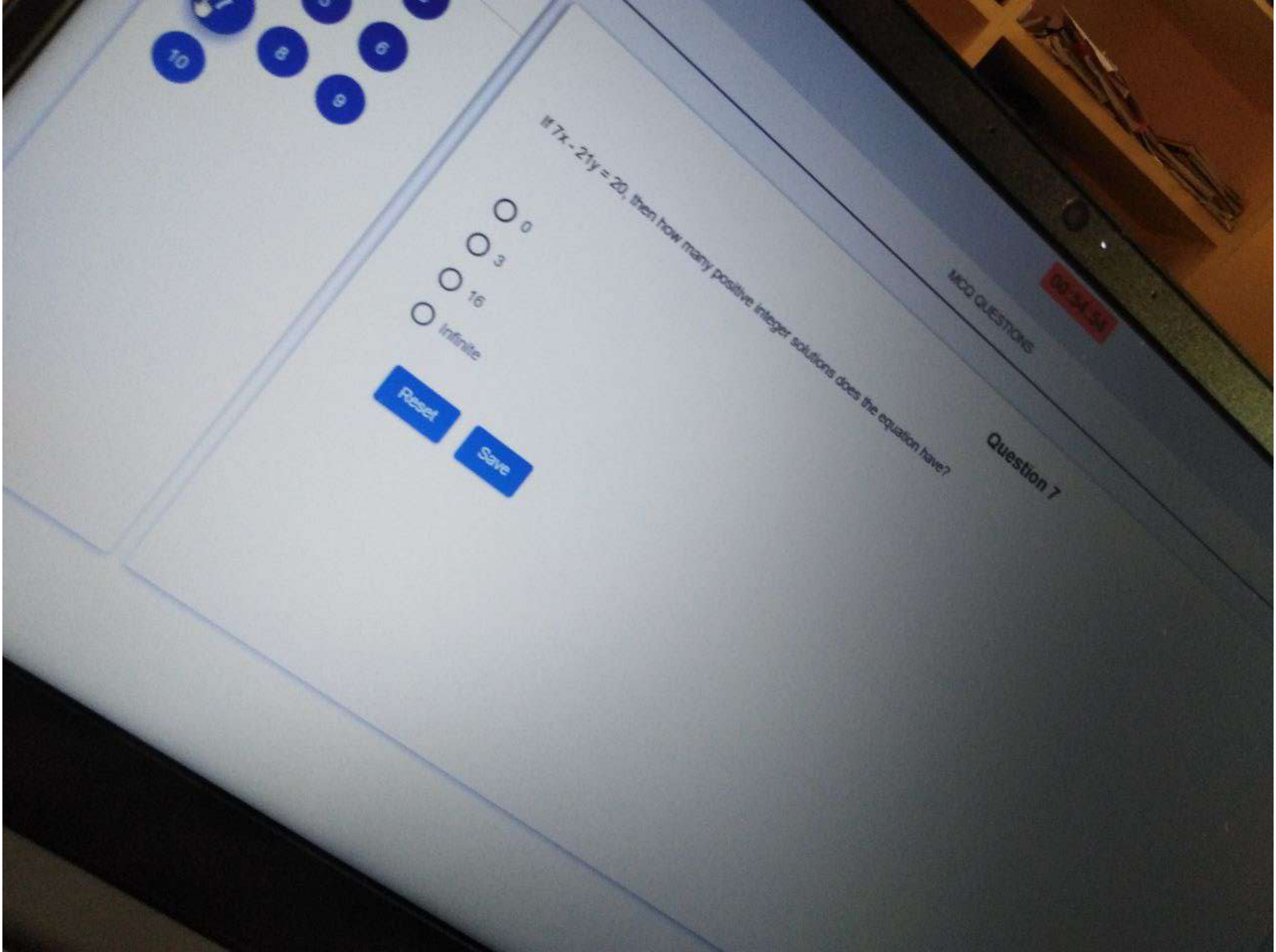
Which among the above solutions would be the best option to implement the above requirement?

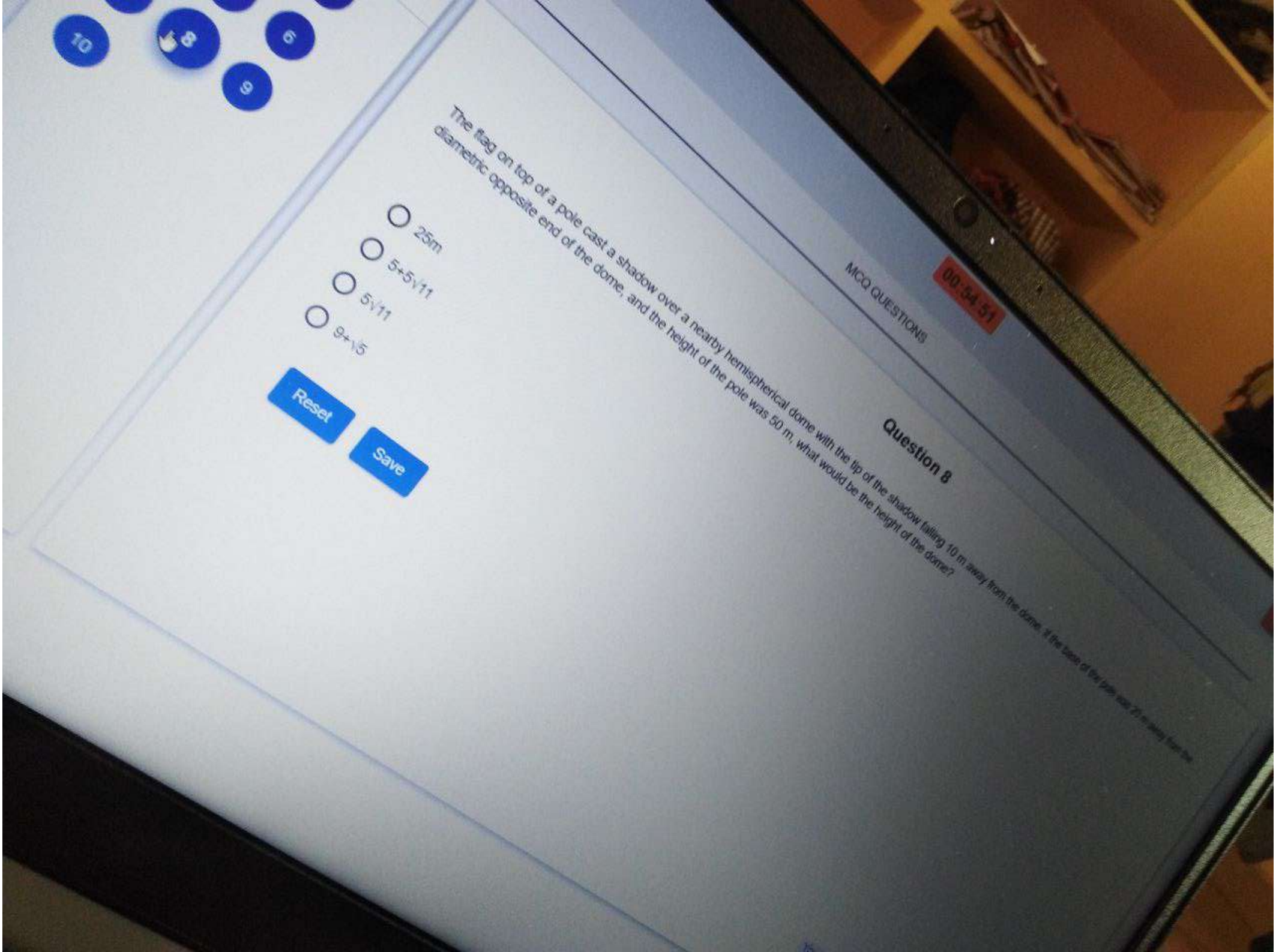
- Solution-1
- Solution-2
- Solution-3
- All three solutions are equally effective

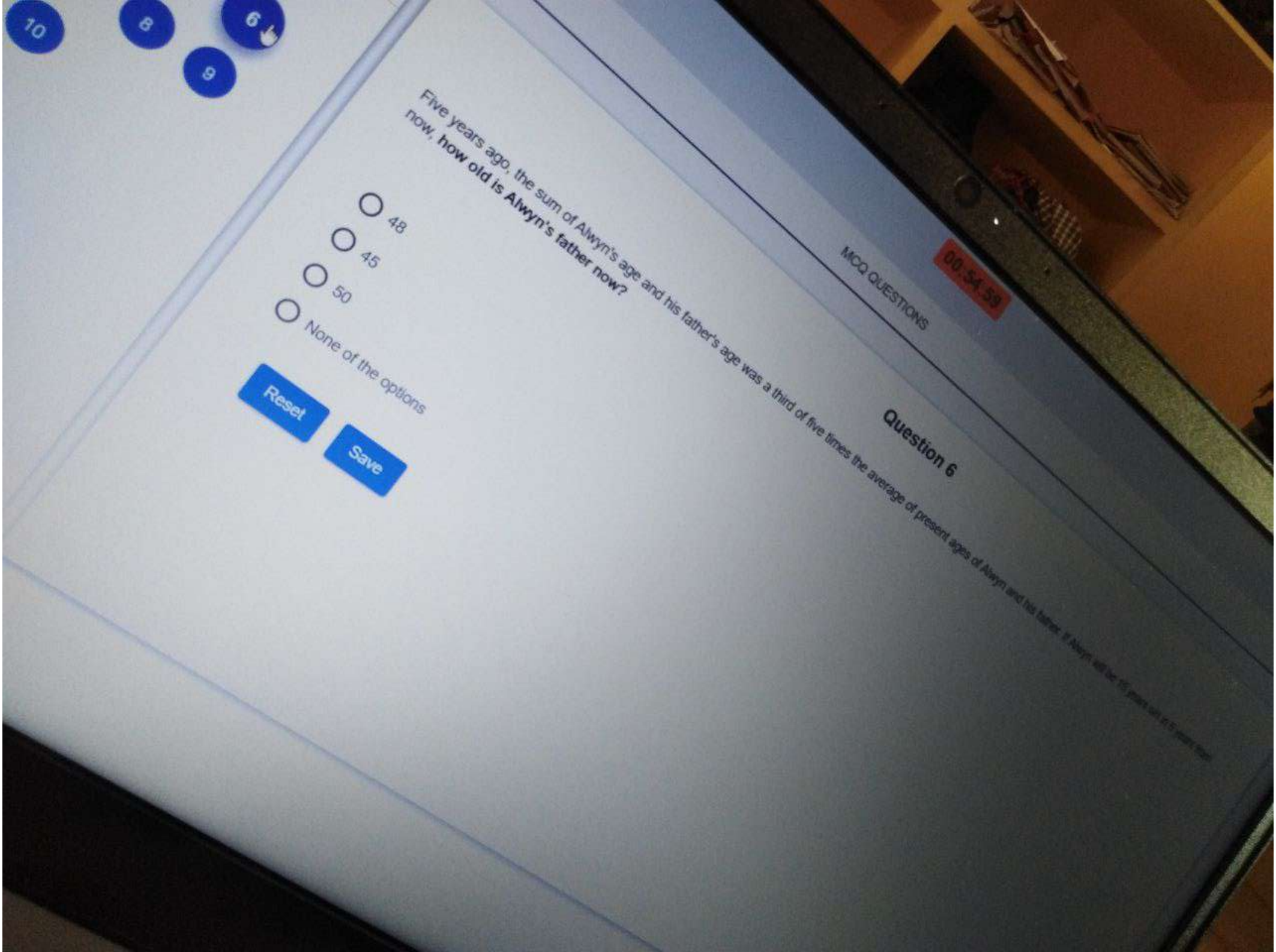
Reset Save

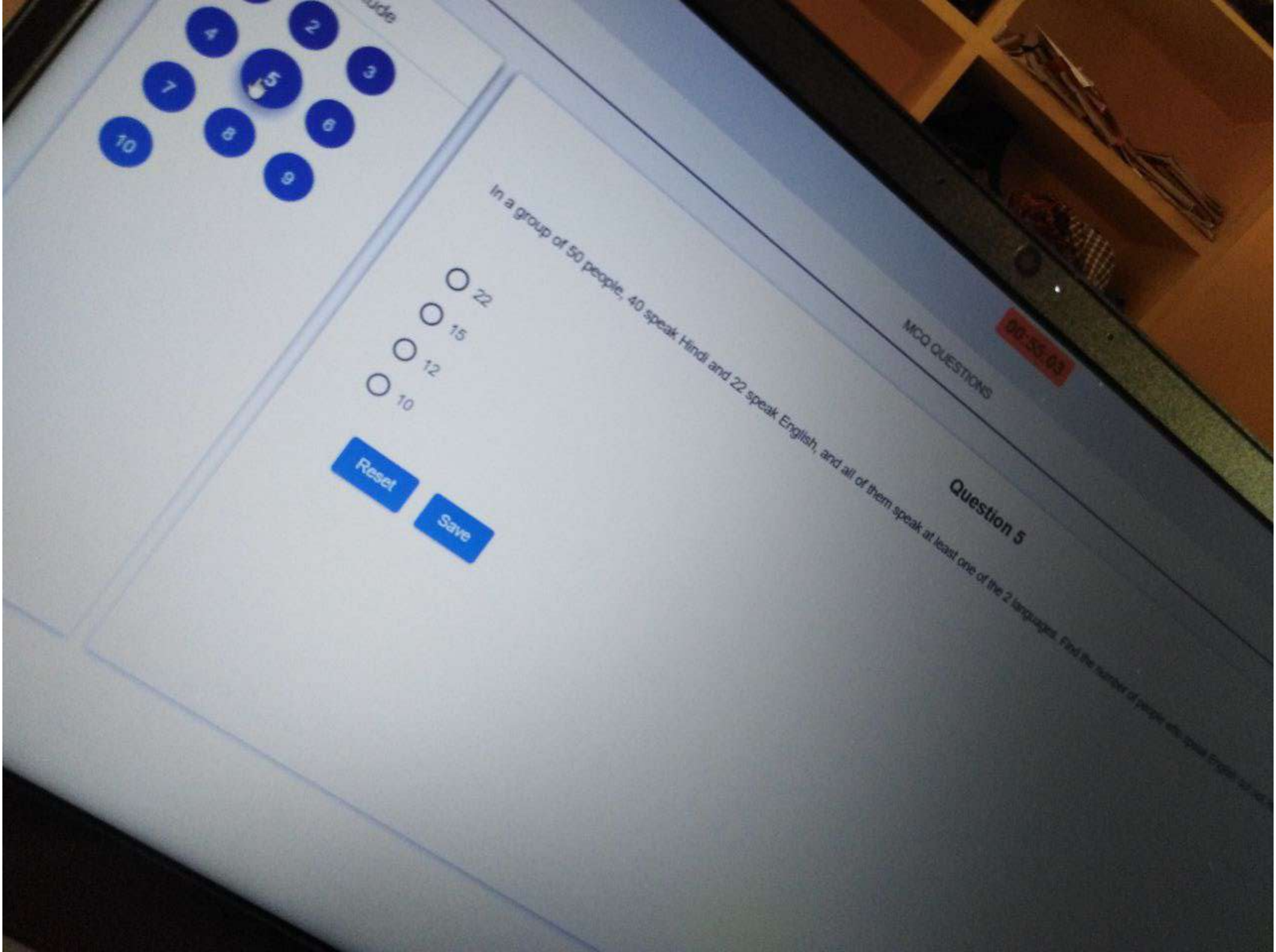
You're being proctored!

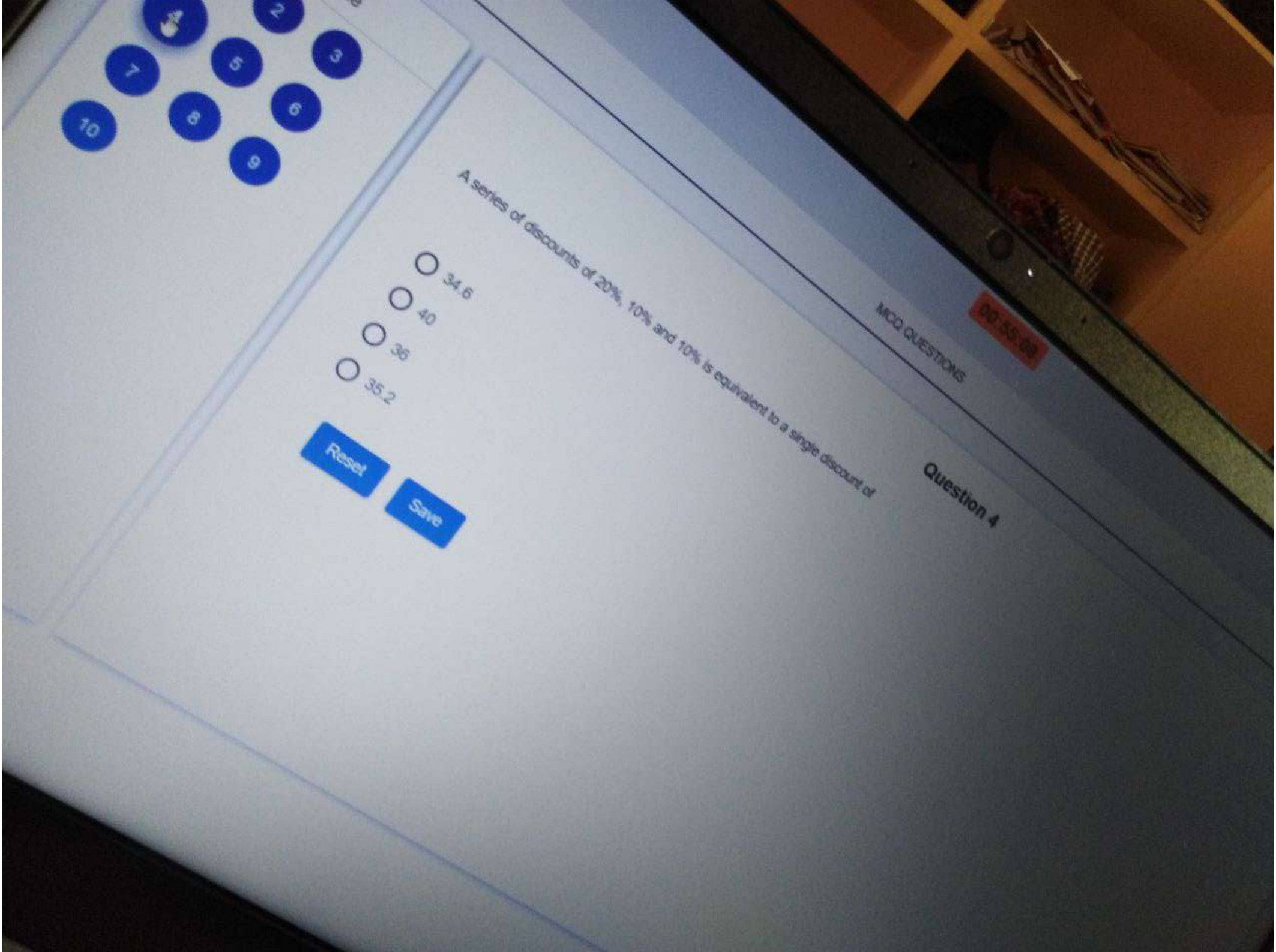

10
8
6
9
MCQ QUESTIONS
Question 7
If 7x - 21y = 20, then how many positive integer solutions does the equation have?
0
3
16
Infinite
Reset
Save

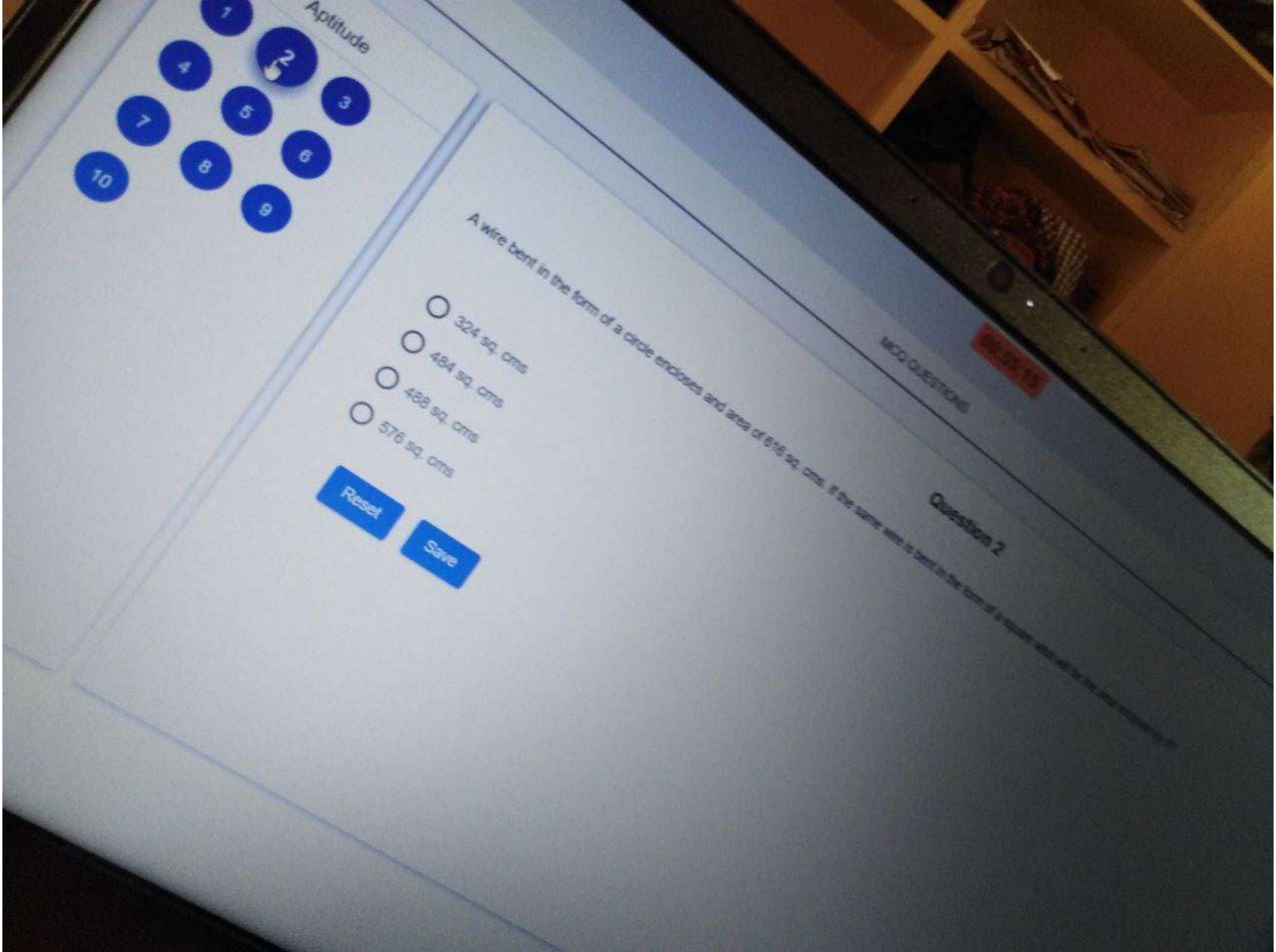

10
8
6
9
MCQ QUESTIONS
Question 7
If 7x - 21y = 20, then how many positive integer solutions does the equation have?
0
3
16
Infinite
Reset
Save

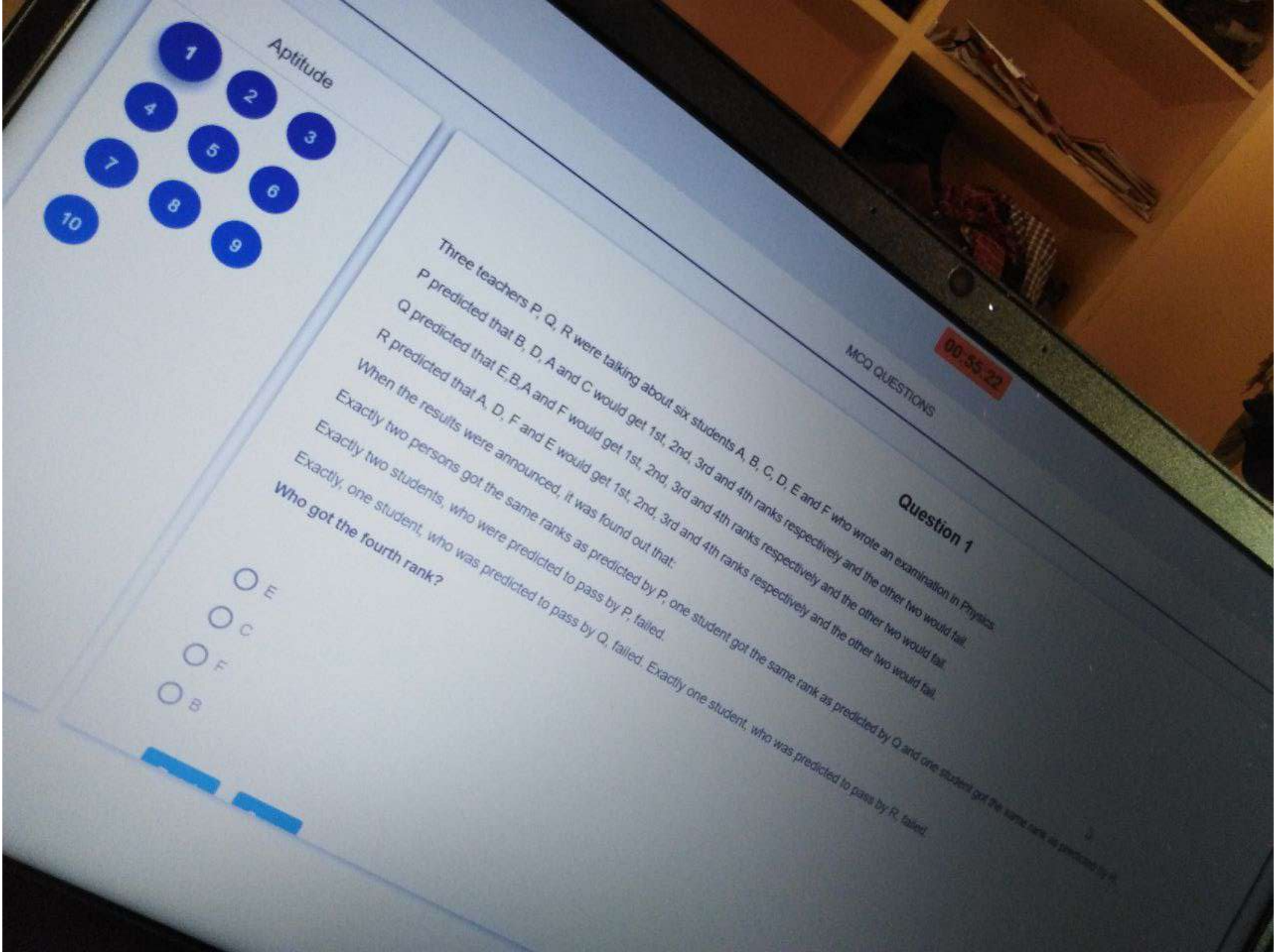

10
8
6
9
MCQ QUESTIONS
00:54:51
Question 8
The flag on top of a pole cast a shadow over a nearby hemispherical dome with the tip of the shadow falling 10 m away from the dome. If the base of the pole was 21 m away from the diametric opposite end of the dome, and the height of the pole was 50 m, what would be the height of the dome?
25m
5+5√11
5√11
9+√5
Reset
Save

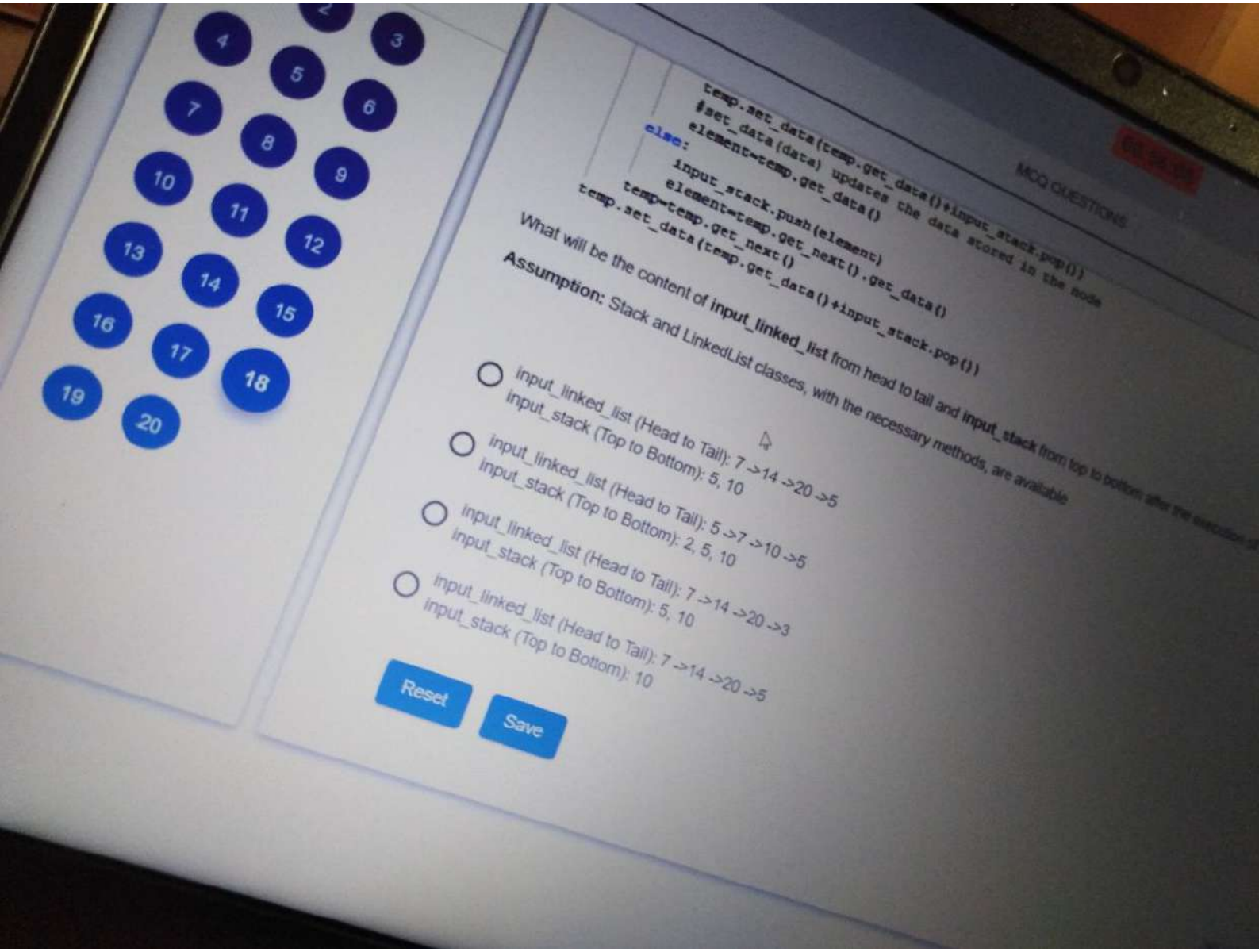

10
8
6
9
MCQ QUESTIONS
00:54:59
Question 6
Five years ago, the sum of Alwyn's age and his father's age was a third of five times the average of present ages of Alwyn and his father. If Alwyn will be 70 years old in 5 years time, now, how old is Alwyn's father now?
48
45
50
None of the options
Reset
Save

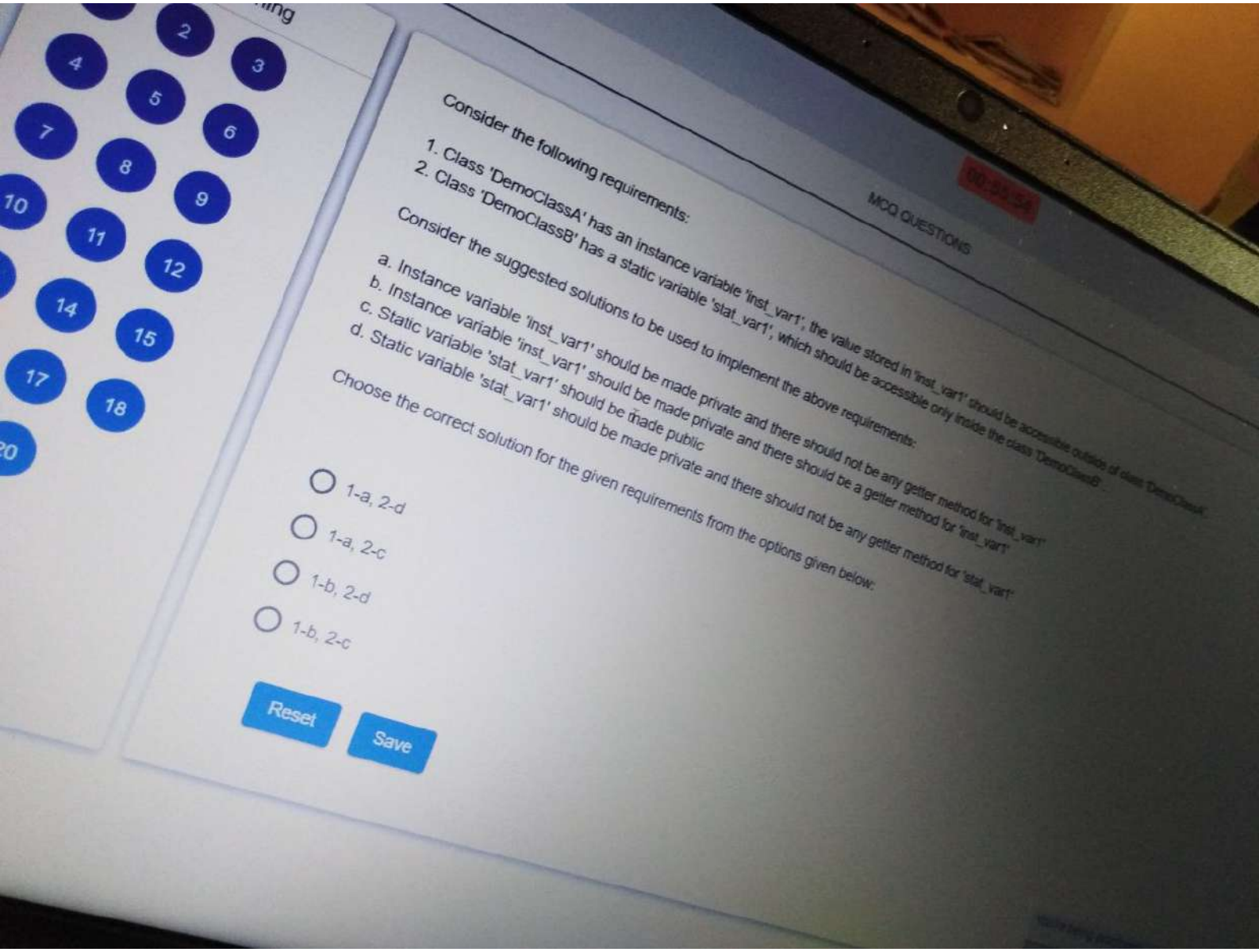

4
2
7
5
3
10
8
6
9
MCQ QUESTIONS
00:35:03
Question 5
In a group of 50 people, 40 speak Hindi and 22 speak English, and all of them speak at least one of the 2 languages. Find the number of people who speak English only.
22
15
12
10
Reset
Save

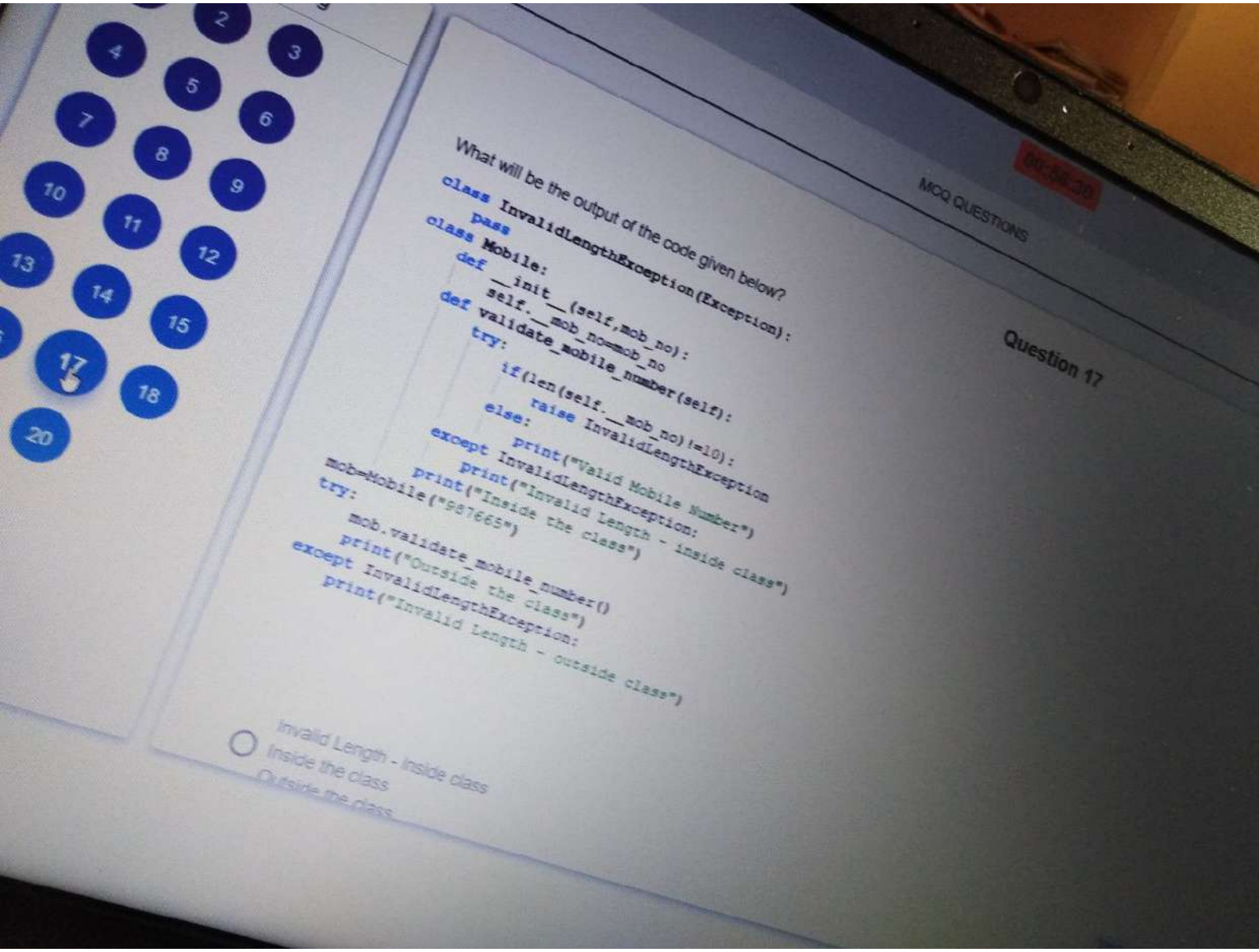

MCQ QUESTIONS
Question 4
A series of discounts of 20%, 10% and 10% is equivalent to a single discount of
34.6
40
36
35.2
Reset
Save

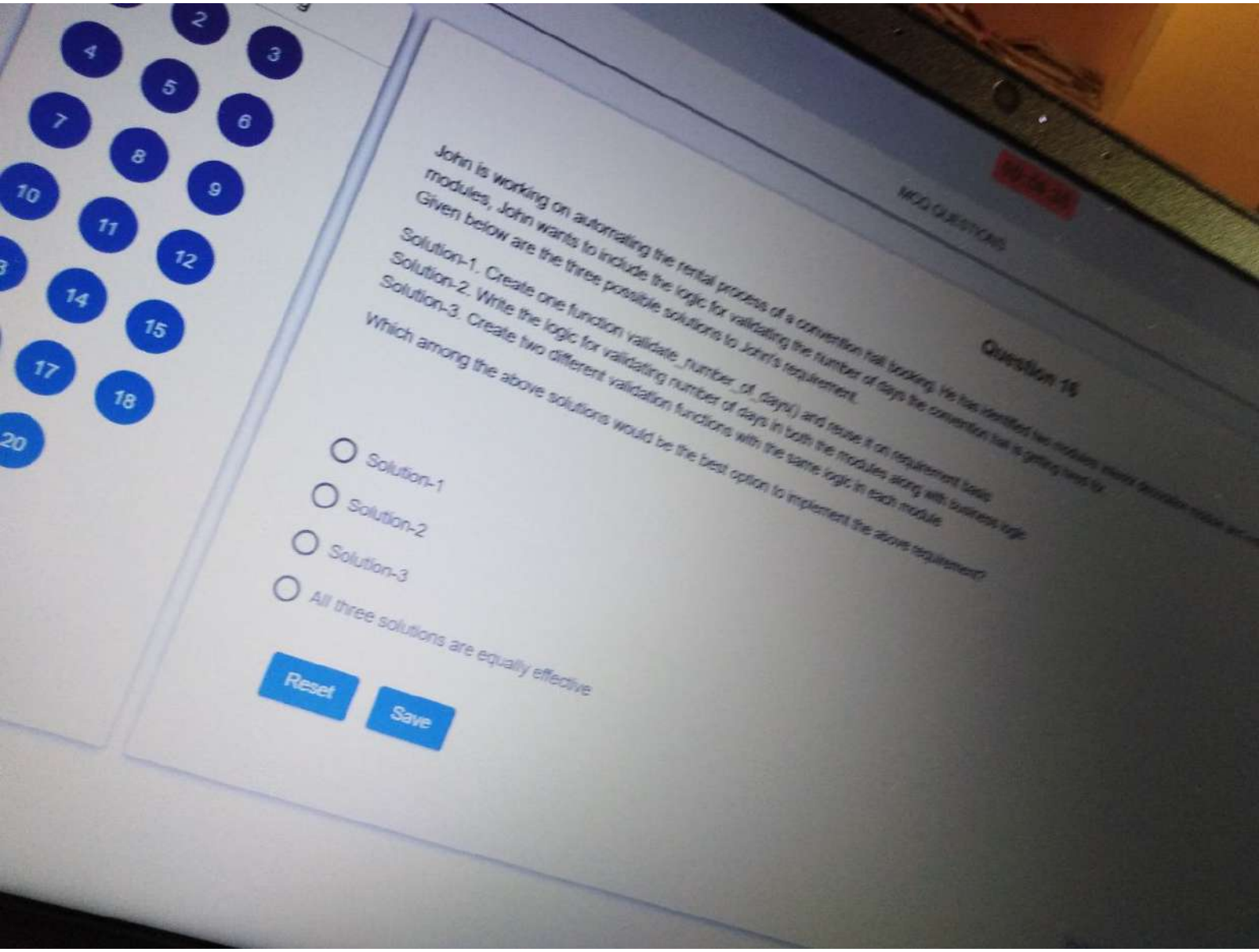

Aptitude
1
2
3
4
5
6
7
8
9
10
MCQ QUESTIONS
Question 2
A wire bent in the form of a circle encloses and area of 616 sq. cms. If the same wire is bent in the form of square what will be the area enclosed
324 sq. cms
484 sq. cms
486 sq. cms
576 sq. cms
Reset
Save

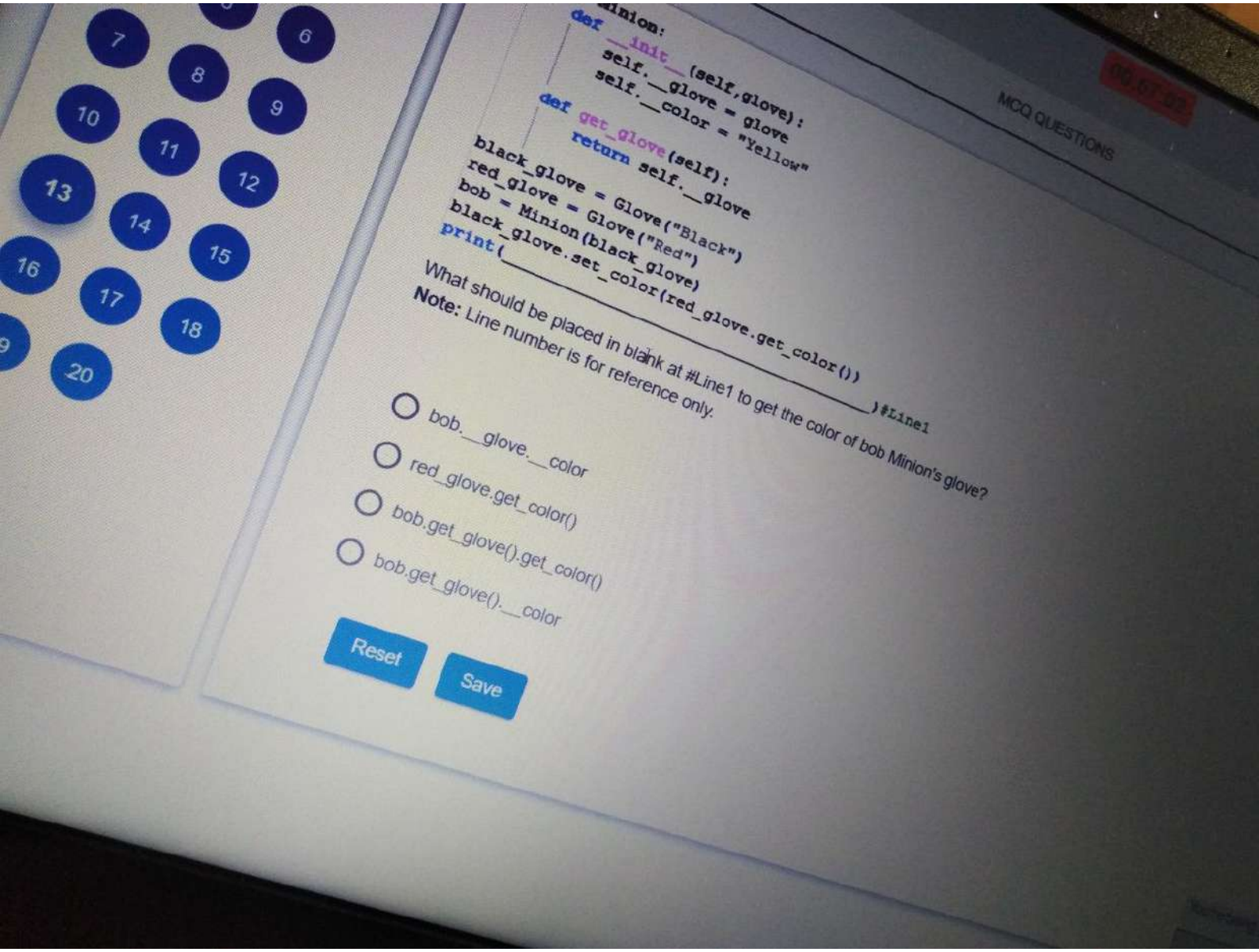

Aptitude
1
2
3
4
5
6
7
8
9
10
MCQ QUESTIONS
00:55:22
Question 1
Three teachers P, Q, R were talking about six students A, B, C, D, E and F who wrote an examination in Physics.
P predicted that B, D, A and C would get 1st, 2nd, 3rd and 4th ranks respectively and the other two would fail.
Q predicted that E,B,A and F would get 1st, 2nd, 3rd and 4th ranks respectively and the other two would fail.
R predicted that A, D, F and E would get 1st, 2nd, 3rd and 4th ranks respectively and the other two would fail.
When the results were announced, it was found out that:
Exactly two persons got the same ranks as predicted by P, one student got the same rank as predicted by Q and one student got the same rank as predicted by R.
Exactly two students, who were predicted to pass by P, failed.
Exactly, one student, who was predicted to pass by Q, failed. Exactly one student, who was predicted to pass by R, failed.
Who got the fourth rank?
E
C
F
B

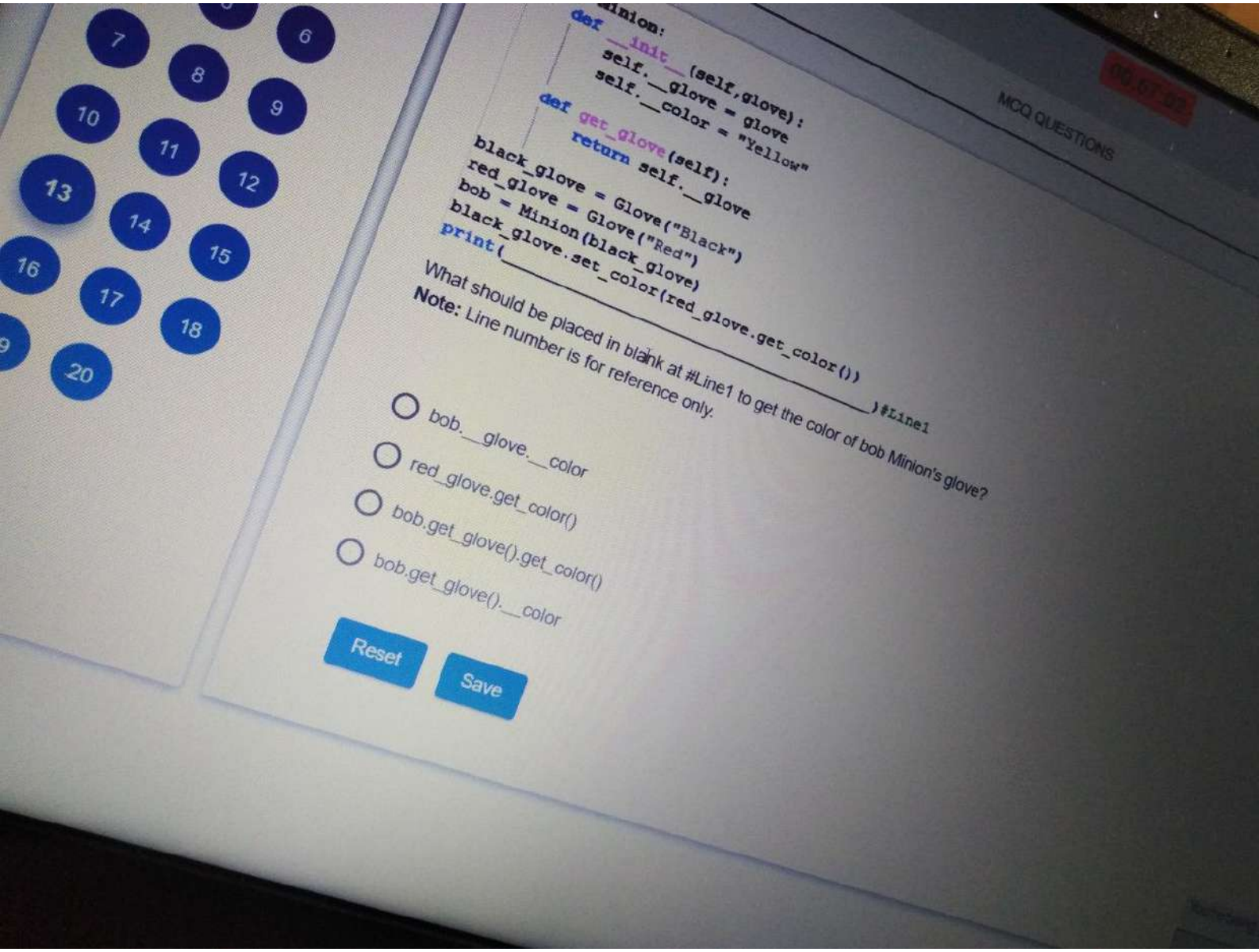

4
3
5
7
6
8
10
9
11
13
12
14
16
15
17
18
19
20
temp.set_data(temp.get_data()+input_stack.pop())
set_data(data) updates the data stored in the node
else:
element=temp.get_data()
input_stack.push(element)
element=temp.get_next().get_data()
temp=temp.get_next()
temp.set_data(temp.get_data()+input_stack.pop())
MCQ QUESTIONS
What will be the content of input_linked_list from head to tail and input_stack from top to bottom after the execution
Assumption: Stack and LinkedList classes, with the necessary methods, are available
input_linked_list (Head to Tail): 7 ->14 ->20 ->5
input_stack (Top to Bottom): 5, 10
input_linked_list (Head to Tail): 5 ->7 ->10 ->5
input_stack (Top to Bottom): 2, 5, 10
input_linked_list (Head to Tail): 7 ->14 ->20 ->3
input_stack (Top to Bottom): 5, 10
input_linked_list (Head to Tail): 7 ->14 ->20 ->5
input_stack (Top to Bottom): 10
Reset
Save

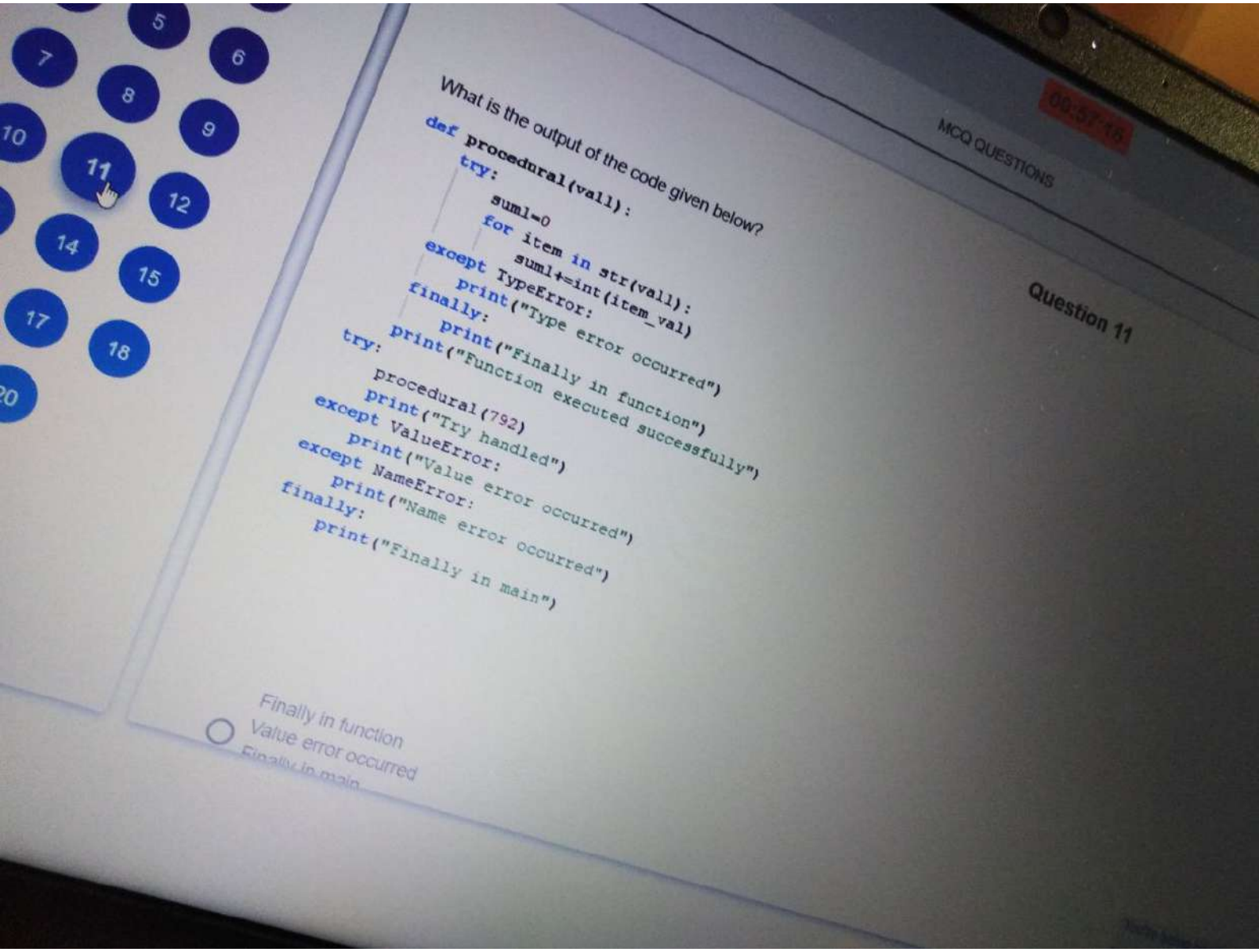

MCQ QUESTIONS
Consider the following requirements:
1. Class 'DemoClassA' has an instance variable 'inst_var1', the value stored in 'inst_var1' should be accessible outside of class 'DemoClassA'
2. Class 'DemoClassB' has a static variable 'stat_var1', which should be accessible only inside the class 'DemoClassB'
Consider the suggested solutions to be used to implement the above requirements:
a. Instance variable 'inst_var1' should be made private and there should not be any getter method for 'inst_var1'
b. Instance variable 'inst_var1' should be made private and there should be a getter method for 'inst_var1'
c. Static variable 'stat_var1' should be made public
d. Static variable 'stat_var1' should be made private and there should not be any getter method for 'stat_var1'
Choose the correct solution for the given requirements from the options given below:
1-a, 2-d
1-a, 2-c
1-b, 2-d
1-b, 2-c
Reset
Save

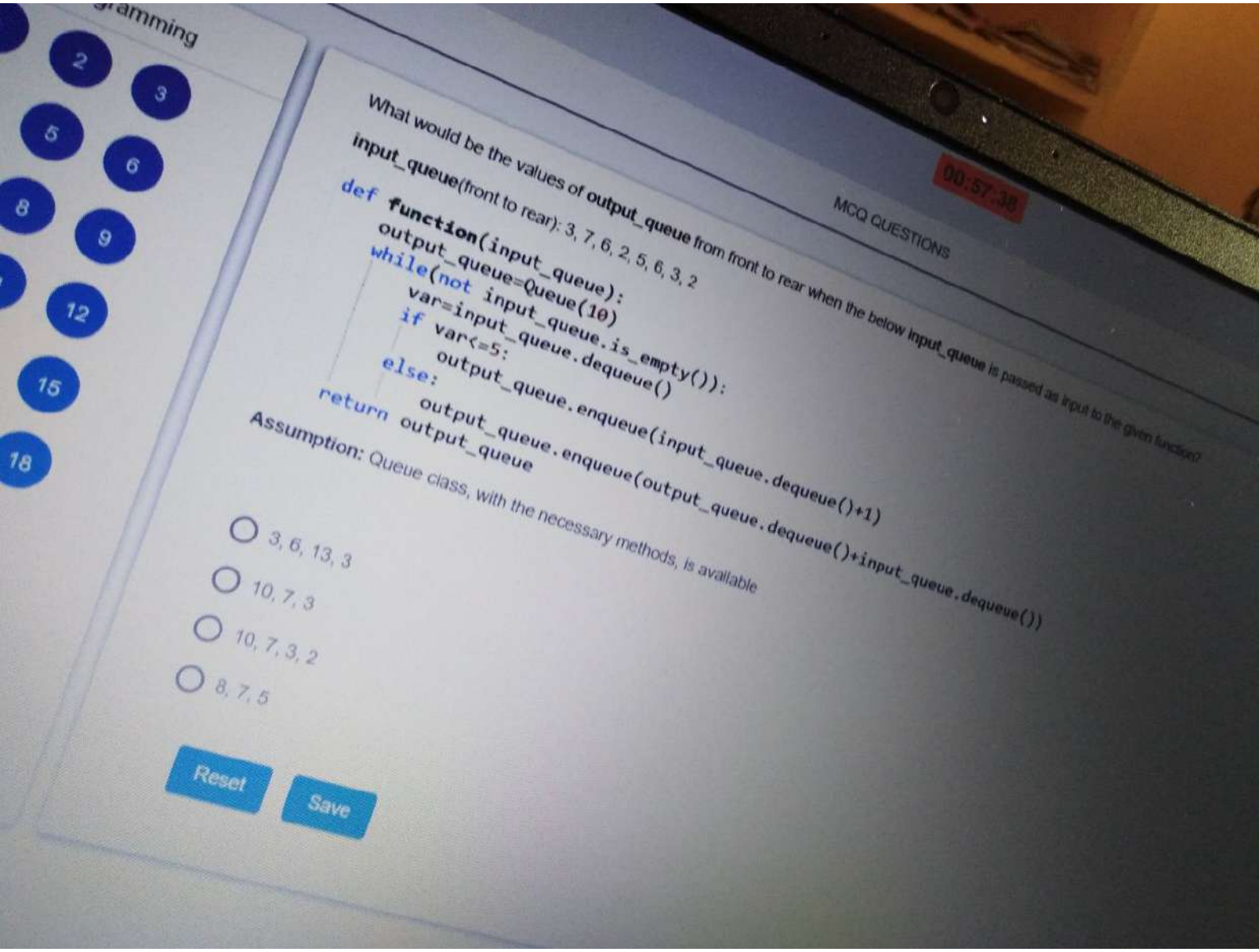

MCQ QUESTIONS
Question 17
What will be the output of the code given below?
class InvalidLengthException(Exception):
pass
class Mobile:
def __init__(self,mob_no):
self.__mob_no=mob_no
def validate_mobile_number(self):
try:
if(len(self.__mob_no)!=10):
raise InvalidLengthException
else:
print("Valid Mobile Number")
except InvalidLengthException:
print("Invalid Length - inside class")
print("Inside the class")
mob=Mobile("987665")
try:
mob.validate_mobile_number()
print("Outside the class")
except InvalidLengthException:
print("Invalid Length - outside class")
Invalid Length - inside class
Inside the class
Outside the class

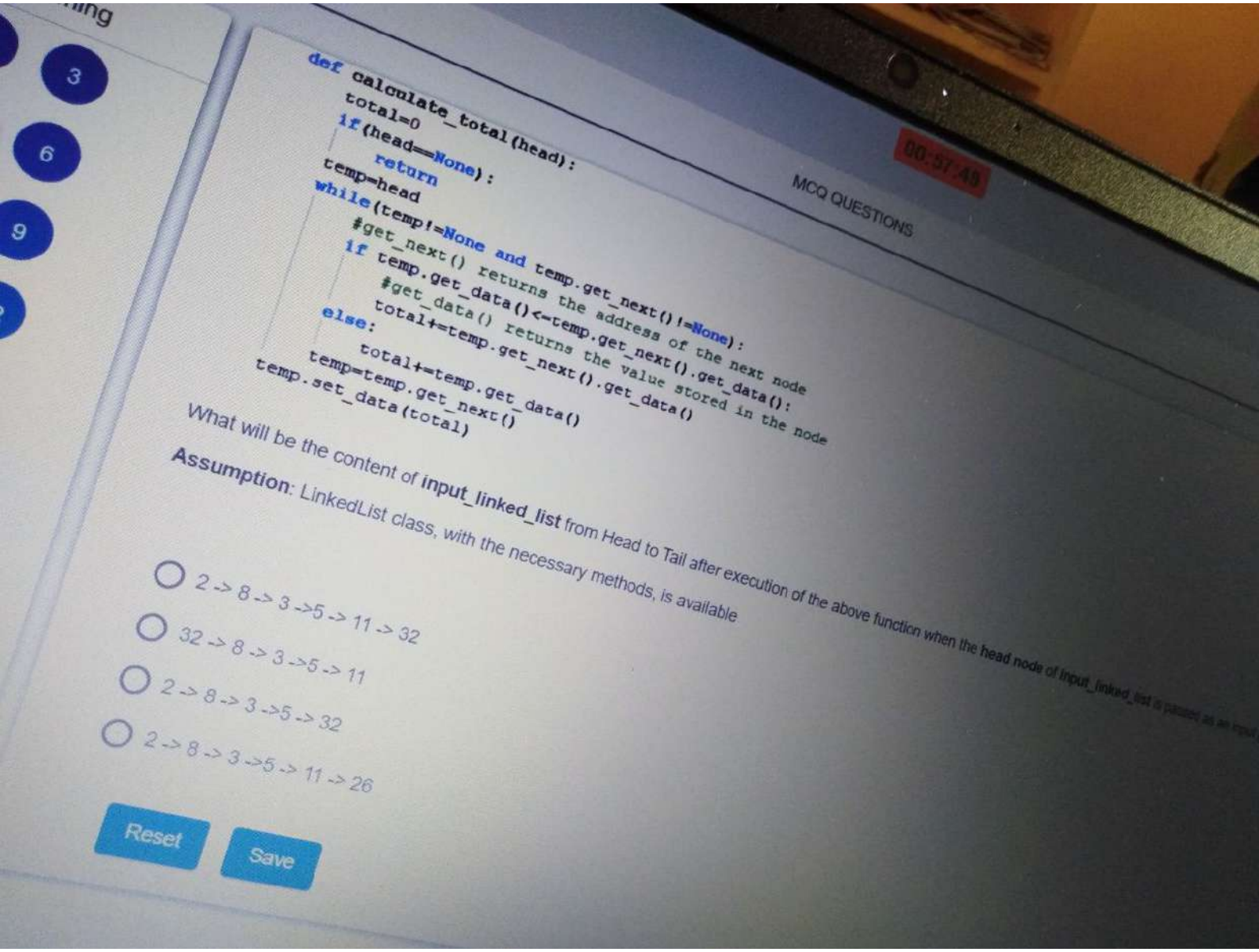

2
4
3
5
7
6
8
10
9
11
12
14
15
17
18
20
MCQ QUESTIONS
Question 15
John is working on automating the rental process of a convention hall booking. He has identified two modules where he can
modules, John wants to include the logic for validating the number of days the convention hall is going to be
Given below are the three possible solutions to John's requirement.
Solution-1. Create one function validate_number_of_days() and reuse it on requirement basis
Solution-2. Write the logic for validating number of days in both the modules along with business logic
Solution-3. Create two different validation functions with the same logic in each module
Which among the above solutions would be the best option to implement the above requirement?
Solution-1
Solution-2
Solution-3
All three solutions are equally effective
Reset
Save

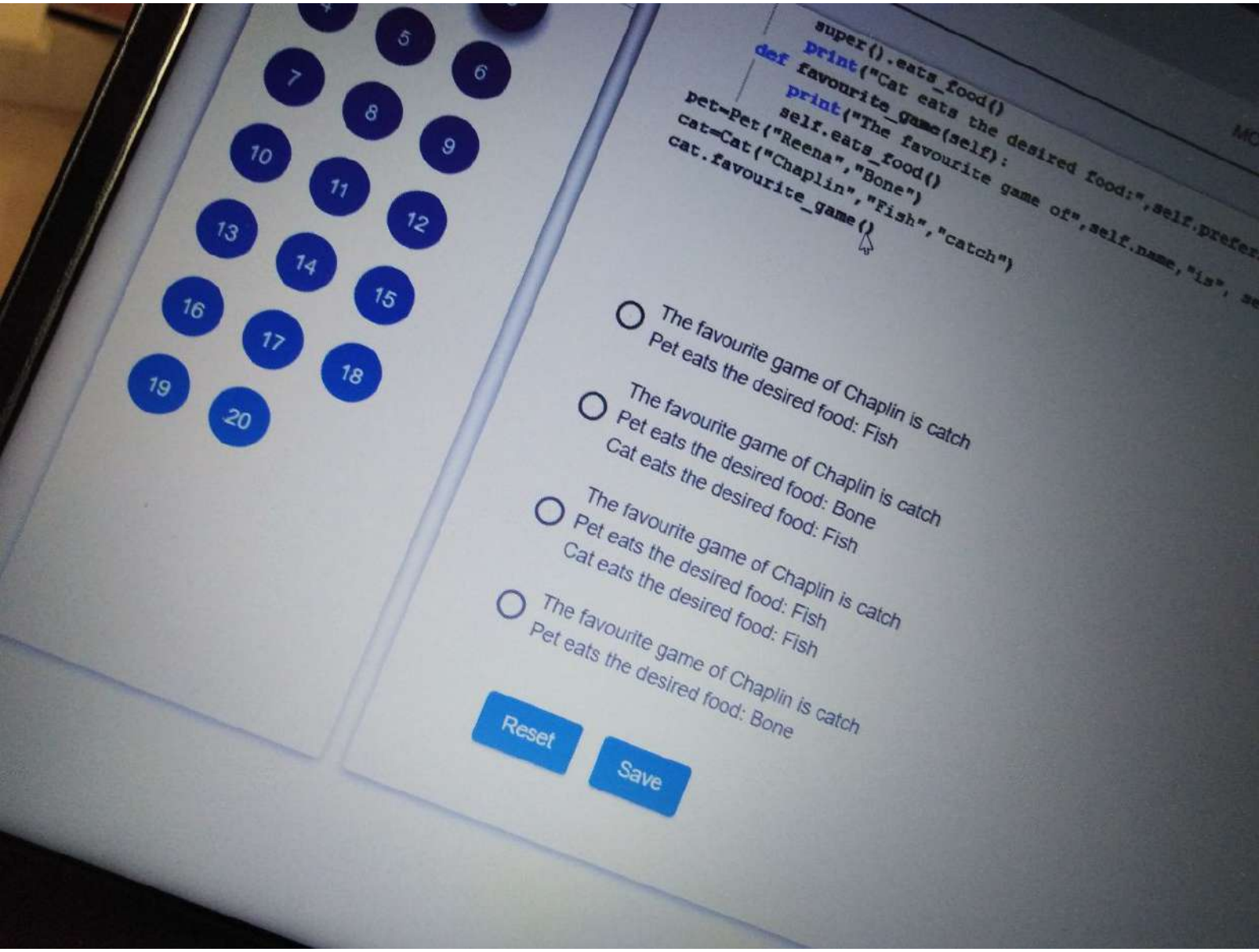

MCQ QUESTIONS
inion:
def __init__(self,glove):
self.__glove = glove
self.__color = "Yellow"
def get_glove(self):
return self.__glove
black_glove = Glove("Black")
red_glove = Glove("Red")
bob = Minion(black_glove)
black_glove.set_color(red_glove.get_color())
print(_____)#Line1
What should be placed in blank at #Line1 to get the color of bob Minion's glove?
Note: Line number is for reference only.
bob.__glove.__color
red_glove.get_color()
bob.get_glove().get_color()
bob.get_glove().__color
Reset
Save

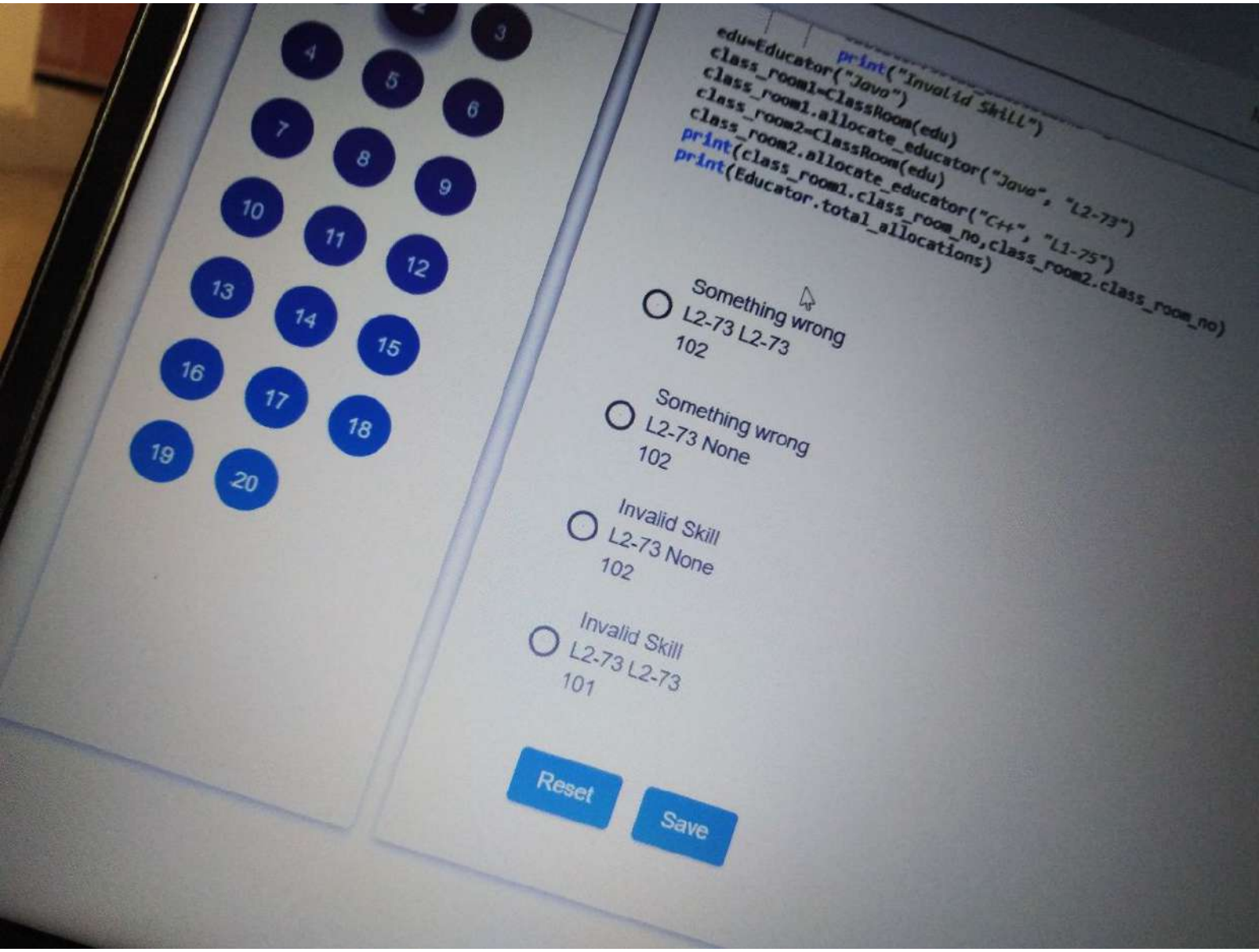

MCQ QUESTIONS
inion:
def __init__(self,glove):
self.__glove = glove
self.__color = "Yellow"
def get_glove(self):
return self.__glove
black_glove = Glove("Black")
red_glove = Glove("Red")
bob = Minion(black_glove)
black_glove.set_color(red_glove.get_color())
print(_____)#Line1
What should be placed in blank at #Line1 to get the color of bob Minion's glove?
Note: Line number is for reference only.
bob.__glove.__color
red_glove.get_color()
bob.get_glove().get_color()
bob.get_glove().__color
Reset
Save

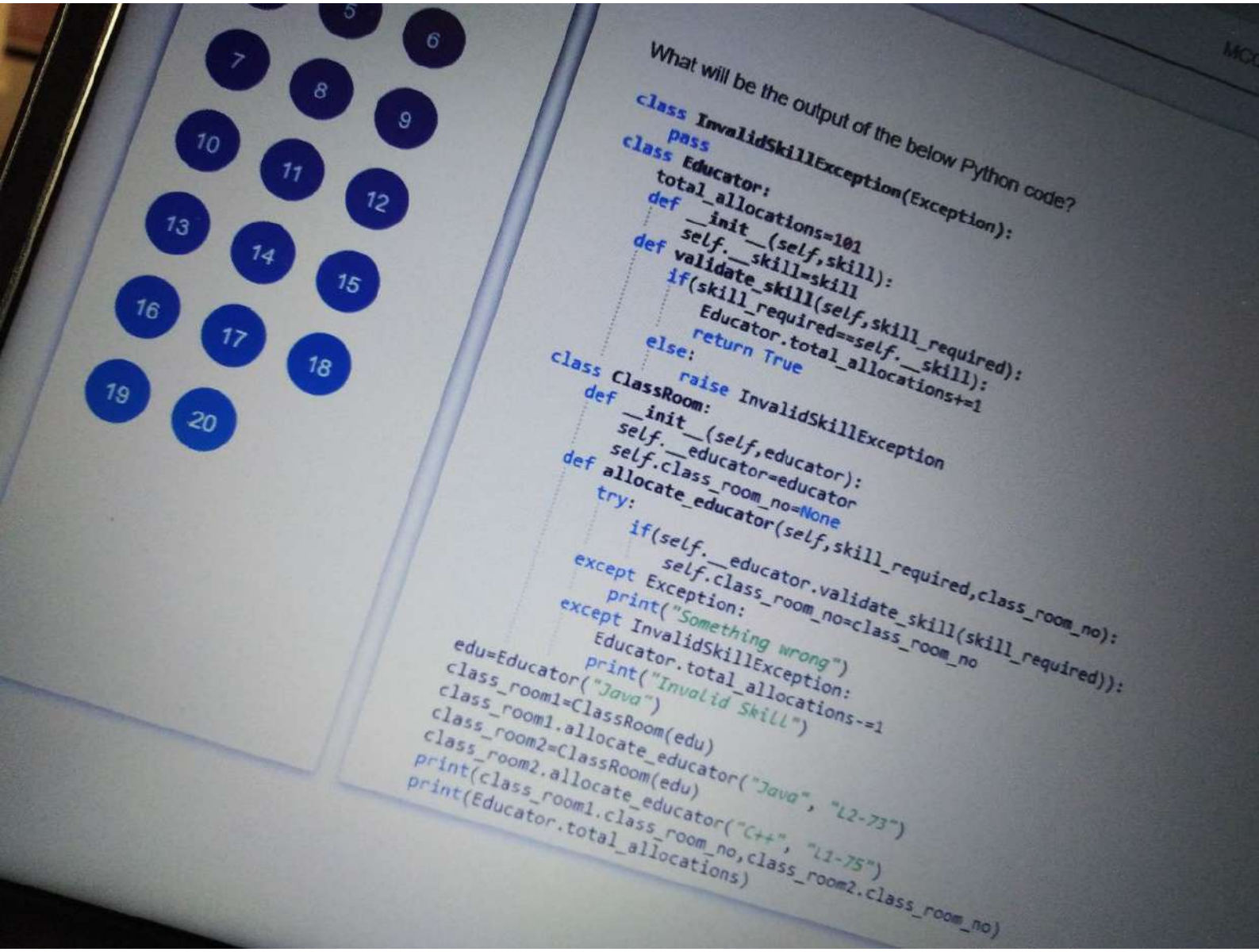

MCQ QUESTIONS
Question 11
What is the output of the code given below?
def procedural(val1):
try:
sum1=0
for item in str(val1):
sum1+=int(item_val)
except TypeError:
print("Type error occurred")
finally:
print("Finally in function")
print("Function executed successfully")
try:
procedural(792)
print("Try handled")
except ValueError:
print("Value error occurred")
except NameError:
print("Name error occurred")
finally:
print("Finally in main")
Finally in function
Value error occurred


amming
2
3
5
6
8
9
12
15
18
MCQ QUESTIONS
00:57:38
What would be the values of output_queue from front to rear when the below input_queue is passed as input to the given function?
input_queue(front to rear): 3, 7, 6, 2, 5, 6, 3, 2
def function(input_queue):
output_queue=Queue(10)
while(not input_queue.is_empty()):
var=input_queue.dequeue()
if var<=5:
output_queue.enqueue(input_queue.dequeue()+1)
else:
output_queue.enqueue(output_queue.dequeue()+input_queue.dequeue())
return output_queue
Assumption: Queue class, with the necessary methods, is available
3, 6, 13, 3
10, 7, 3
10, 7, 3, 2
8, 7, 5
Reset
Save


MCQ QUESTIONS
00:57:49
def calculate_total(head):
total=0
if(head==None):
return
temp=head
while(temp!=None and temp.get_next()!=None):
#get_next() returns the address of the next node
if temp.get_data()<=temp.get_next().get_data():
#get_data() returns the value stored in the node
total+=temp.get_data()
else:
total+=temp.get_data()
temp=temp.get_next()
temp.set_data(total)
What will be the content of input_linked_list from Head to Tail after execution of the above function when the head node of input_linked_list is passed as an input
Assumption: LinkedList class, with the necessary methods, is available
2 -> 8 -> 3 ->5 -> 11 -> 32
32 -> 8 -> 3 ->5 -> 11
2 -> 8 -> 3 ->5 -> 32
2 -> 8 -> 3 ->5 -> 11 -> 26
Reset
Save


5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
super().eats_food()
print("Cat eats the desired food:",self.prefer
def favourite_game(self):
print("The favourite game of",self.name,"is",
self.eats_food()
pet=Pet("Reena","Bone")
cat=Cat("Chaplin","Fish","catch")
cat.favourite_game()
The favourite game of Chaplin is catch
Pet eats the desired food: Fish
The favourite game of Chaplin is catch
Pet eats the desired food: Bone
Cat eats the desired food: Fish
The favourite game of Chaplin is catch
Pet eats the desired food: Fish
Cat eats the desired food: Fish
The favourite game of Chaplin is catch
Pet eats the desired food: Bone
Reset
Save


print("Invalid Skill")
edu=Educator("Java")
class_room1=ClassRoom(edu)
class_room1.allocate_educator("Java", "L2-73")
class_room2=ClassRoom(edu)
class_room2.allocate_educator("C++", "L1-75")
print(class_room1.class_room_no,class_room2.class_room_no)
print(Educator.total_allocations)
Something wrong
L2-73 L2-73
102
Something wrong
L2-73 None
102
Invalid Skill
L2-73 None
102
Invalid Skill
L2-73 L2-73
101
Reset
Save


5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
What will be the output of the below Python code?
class InvalidSkillException(Exception):
pass
class Educator:
total_allocations=101
def __init__(self,skill):
self.__skill=skill
def validate_skill(self,skill_required):
if(skill_required==self.__skill):
Educator.total_allocations+=1
return True
else:
raise InvalidSkillException
class ClassRoom:
def __init__(self,educator):
self.__educator=educator
self.class_room_no=None
def allocate_educator(self,skill_required,class_room_no):
try:
if(self.__educator.validate_skill(skill_required)):
self.class_room_no=class_room_no
except Exception:
print("Something wrong")
except InvalidSkillException:
Educator.total_allocations-=1
print("Invalid Skill")
edu=Educator("Java")
class_room1=ClassRoom(edu)
class_room1.allocate_educator("Java", "L2-73")
class_room2=ClassRoom(edu)
class_room2.allocate_educator("C++", "L1-75")
print(class_room1.class_room_no,class_room2.class_room_no)
print(Educator.total_allocations)

4 5 6

7 8 9

10 11 12

13 14 15

16 17 18

19 20

Ques

What is the output of the code given below?

```
values = ["823", "863"]
num = values[0][0:]
for row in range(0, len(values)):
    for column in range(0, len(values[row])):
        if num > values[row][column:]:
            num = values[row][column:]
print(num)
```

- 2
- 3
- 23
- 823

Reset Save

4 5 6

7 8 9

10 11 12

13 14 15

16 17 18

19 20

Ques

What is the output of the code given below?

```
values = ["823", "863"]
num = values[0][0:]
for row in range(0, len(values)):
    for column in range(0, len(values[row])):
        if num > values[row][column:]:
            num = values[row][column:]
print(num)
```

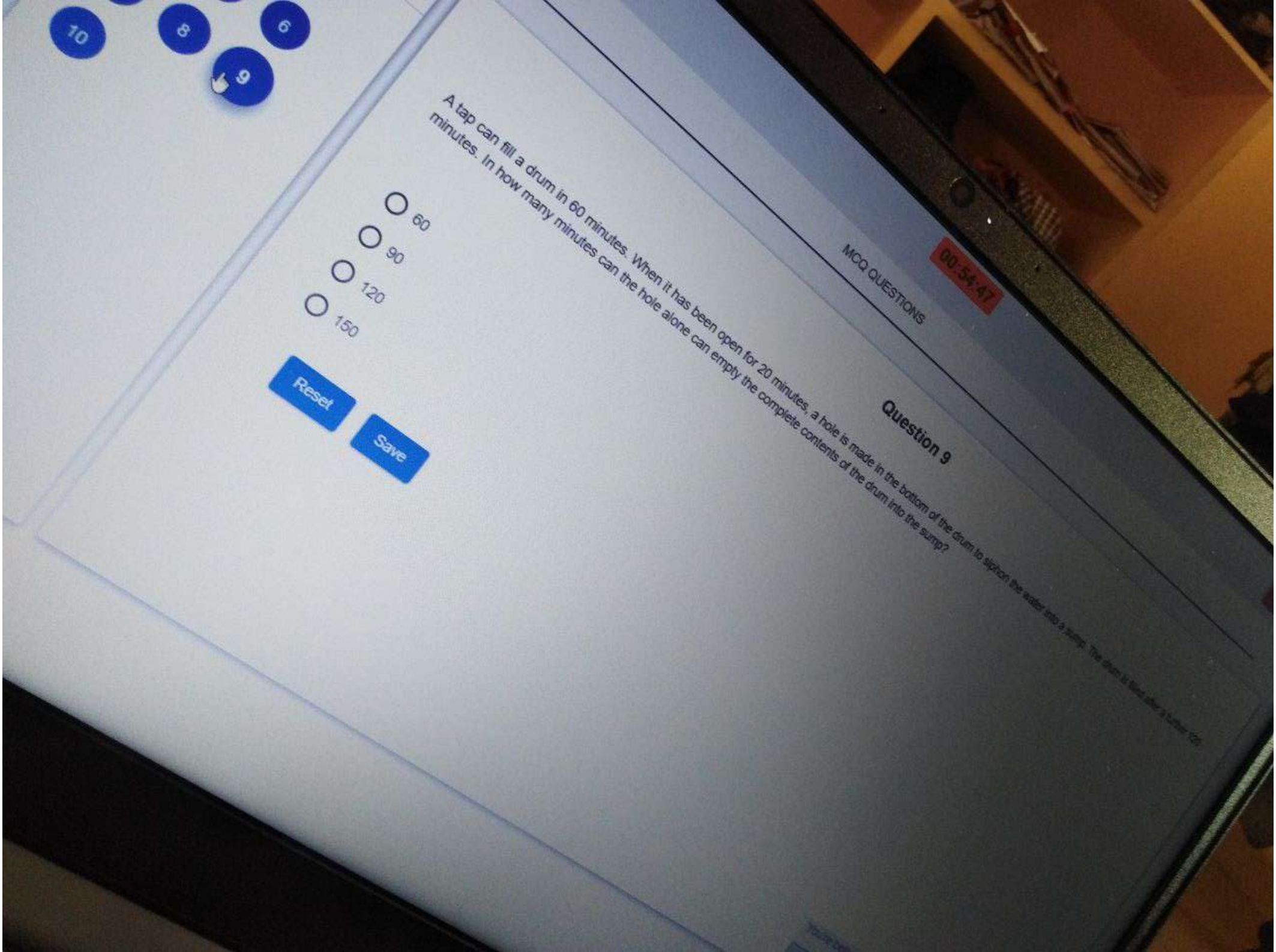
- 2
- 3
- 23
- 823

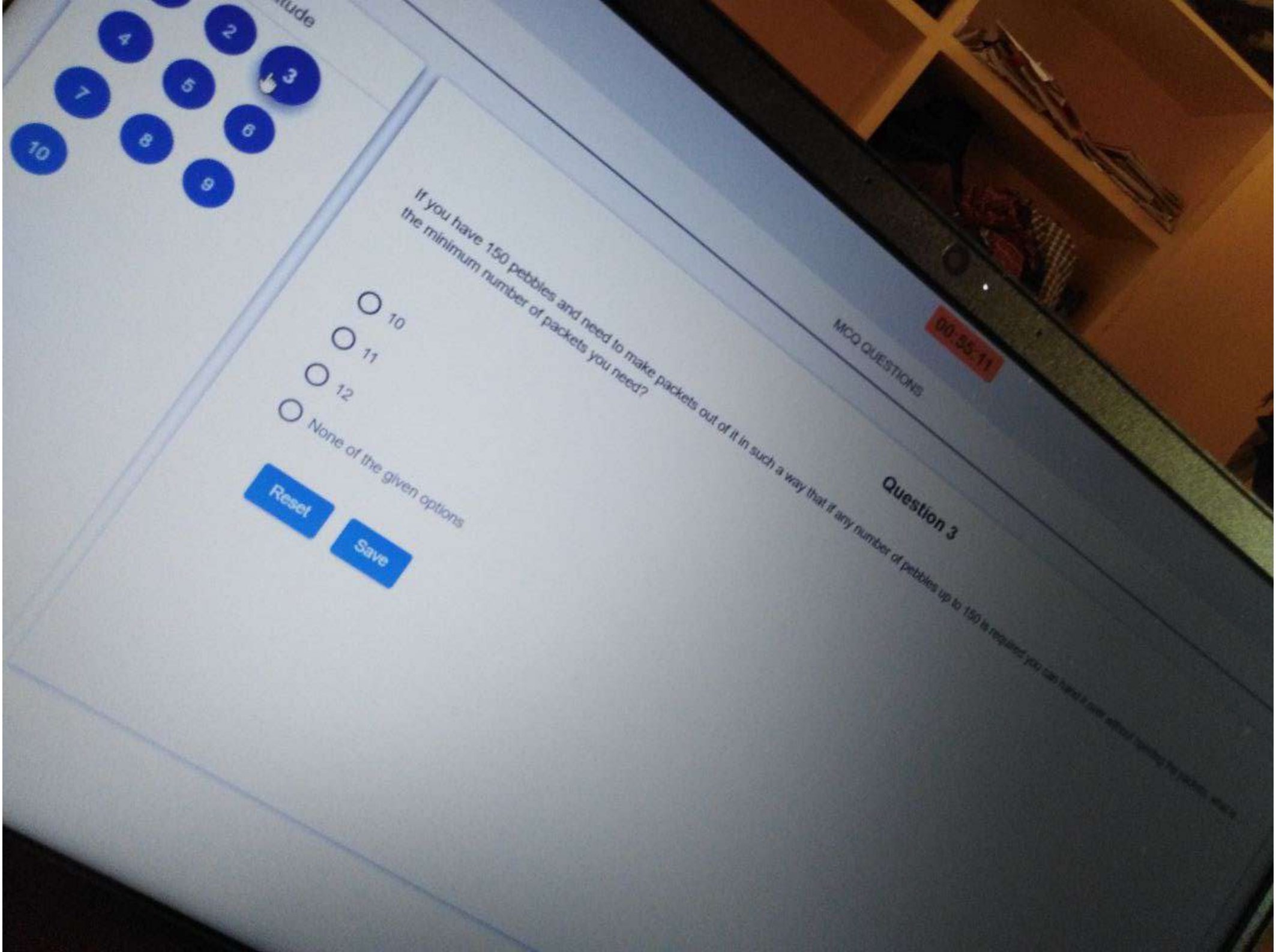
Reset Save

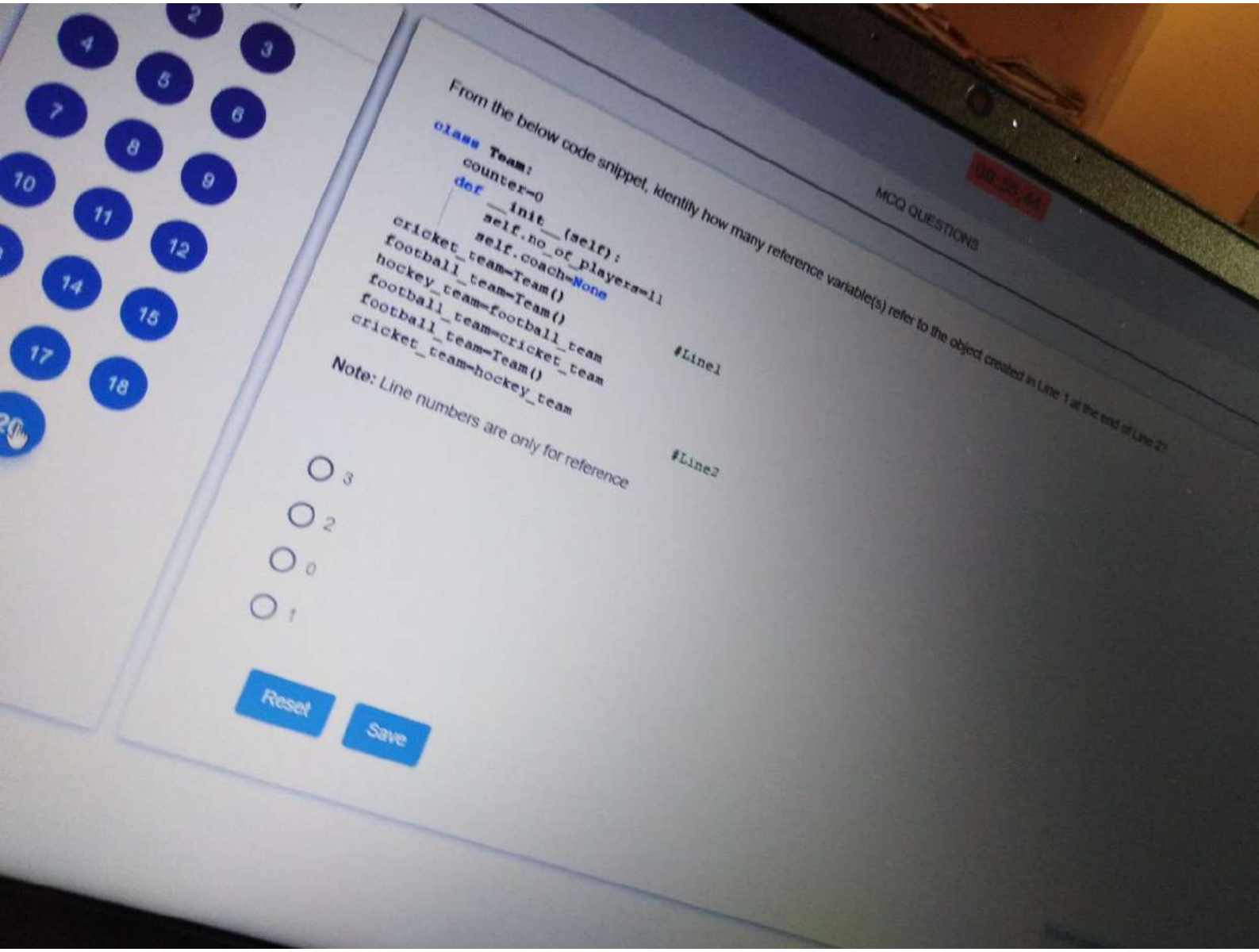
What is the output of the code given below?

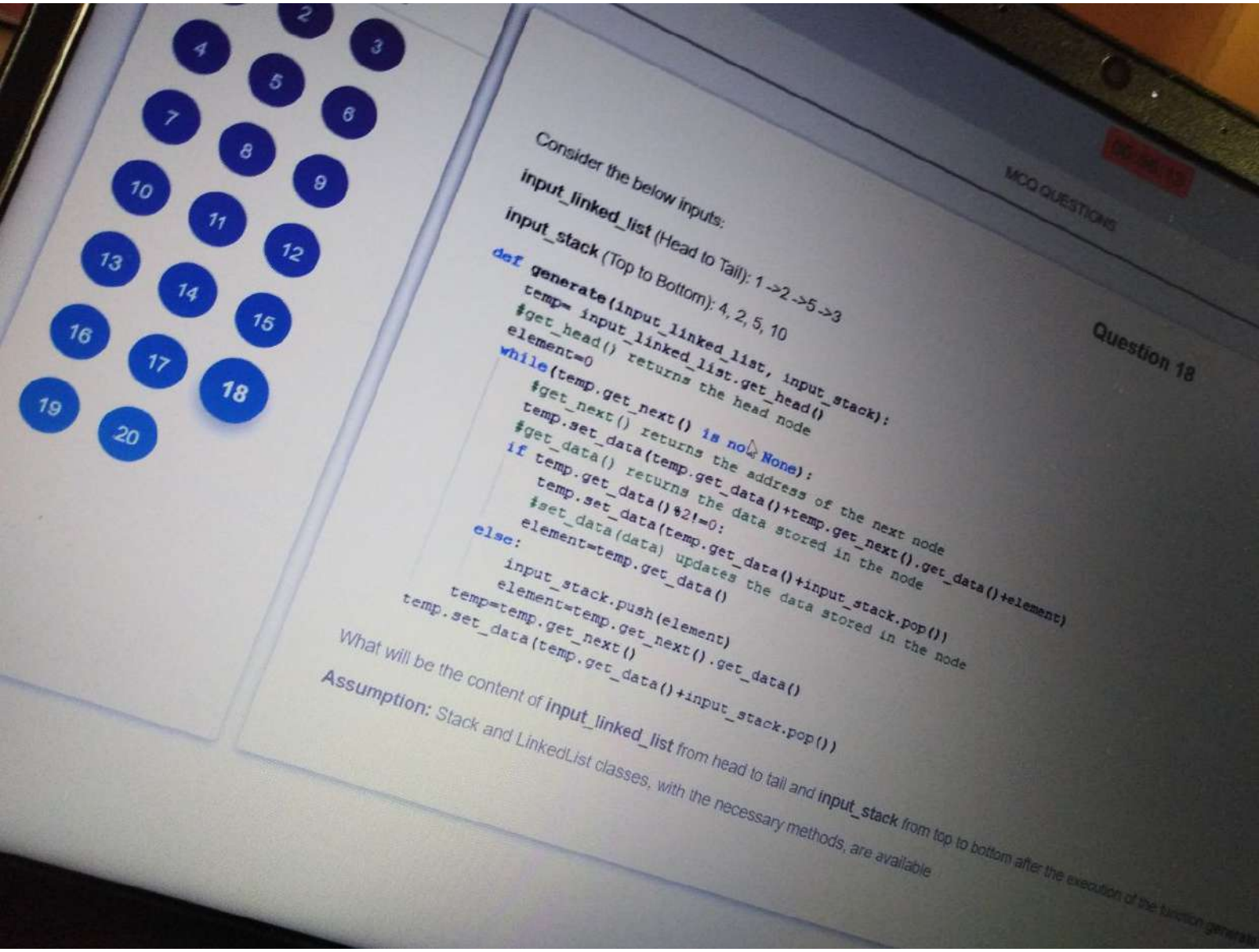
```
values = ["823", "863"]
num = values[0][0:]
for row in range(0, len(values)):
    for column in range(0, len(values[row])):
        if num > values[row][column:]:
            num = values[row][column:]
print(num)
```

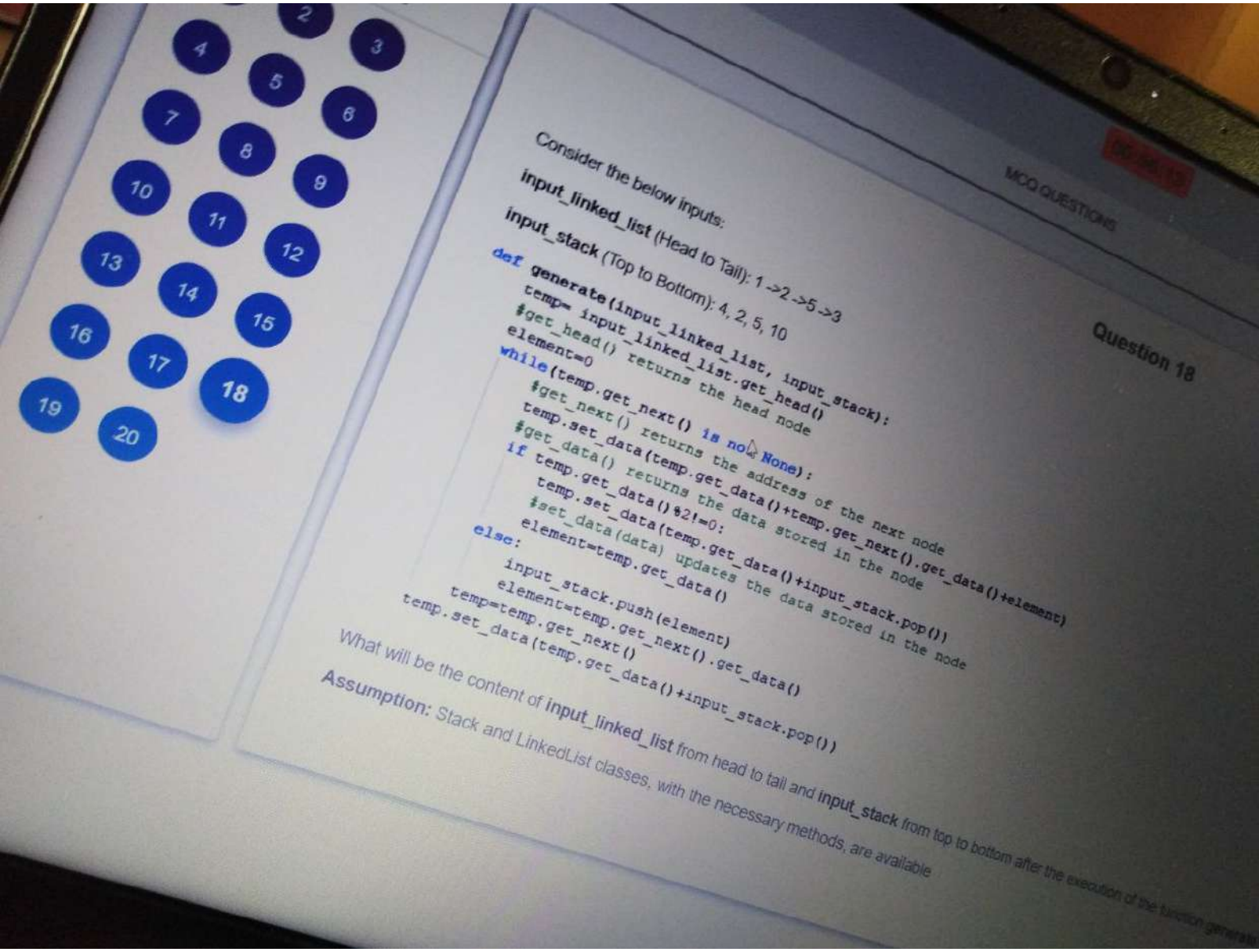
- 2
- 3
- 23
- 823

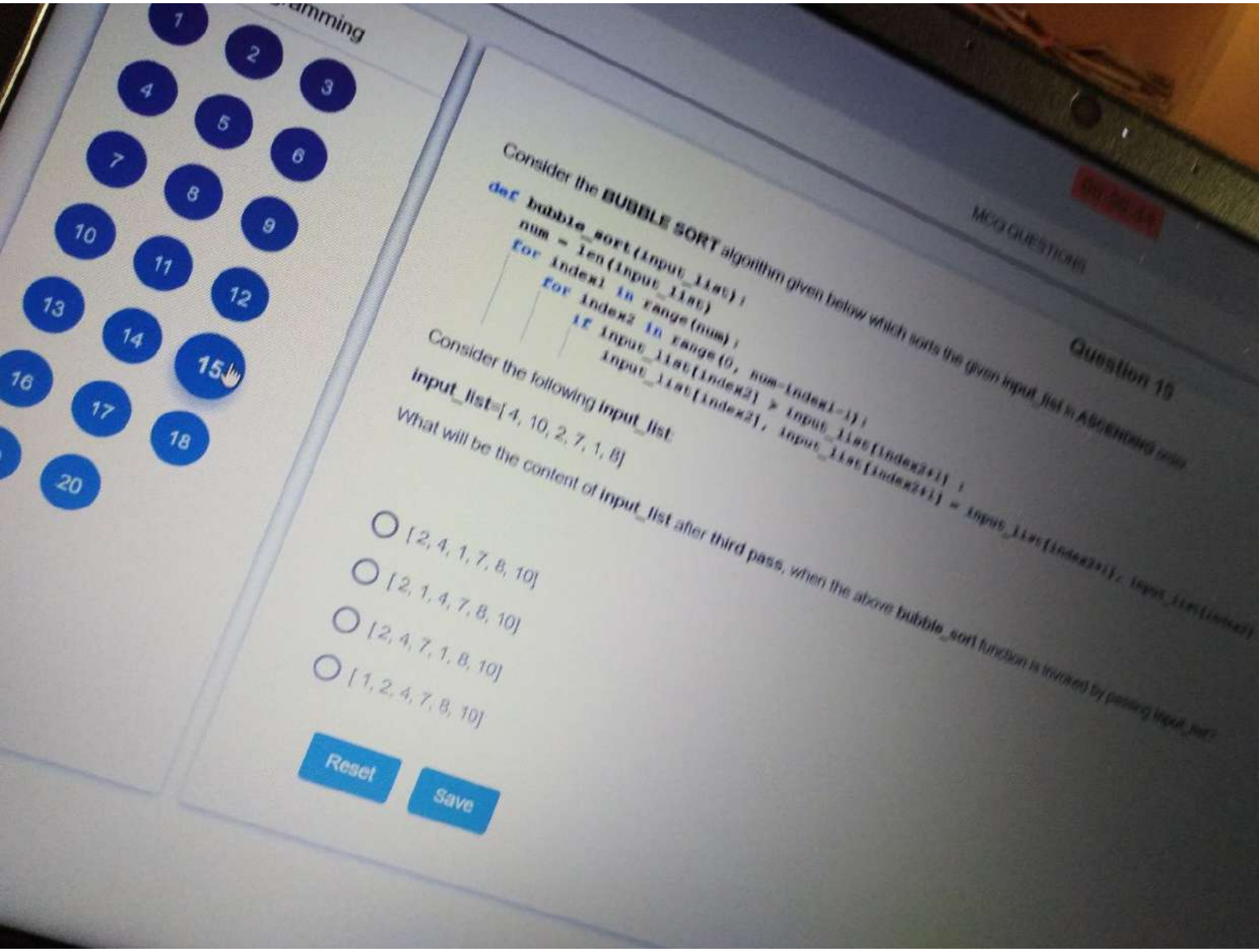

10
8
6
9
MCQ QUESTIONS
00:54:47
Question 9
A tap can fill a drum in 60 minutes. When it has been open for 20 minutes, a hole is made in the bottom of the drum to siphon the water into a sump. The drum is filled after a further 60 minutes. In how many minutes can the hole alone can empty the complete contents of the drum into the sump?
60
90
120
150
Reset
Save

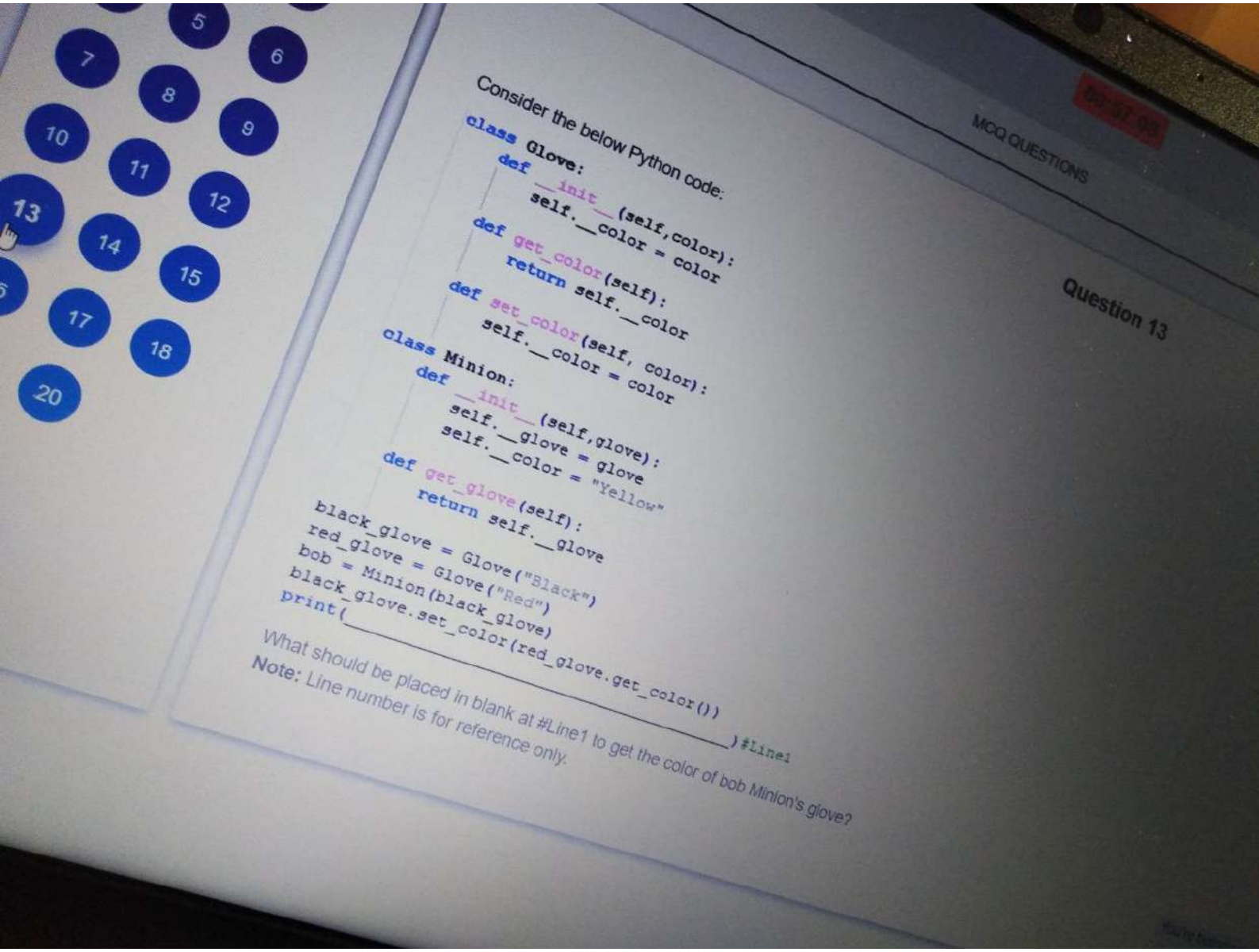

MCQ QUESTIONS
00:55:11
Question 3
If you have 150 pebbles and need to make packets out of it in such a way that if any number of pebbles up to 150 is required you can hand it over without having the packets and
the minimum number of packets you need?
10
11
12
None of the given options
Reset
Save

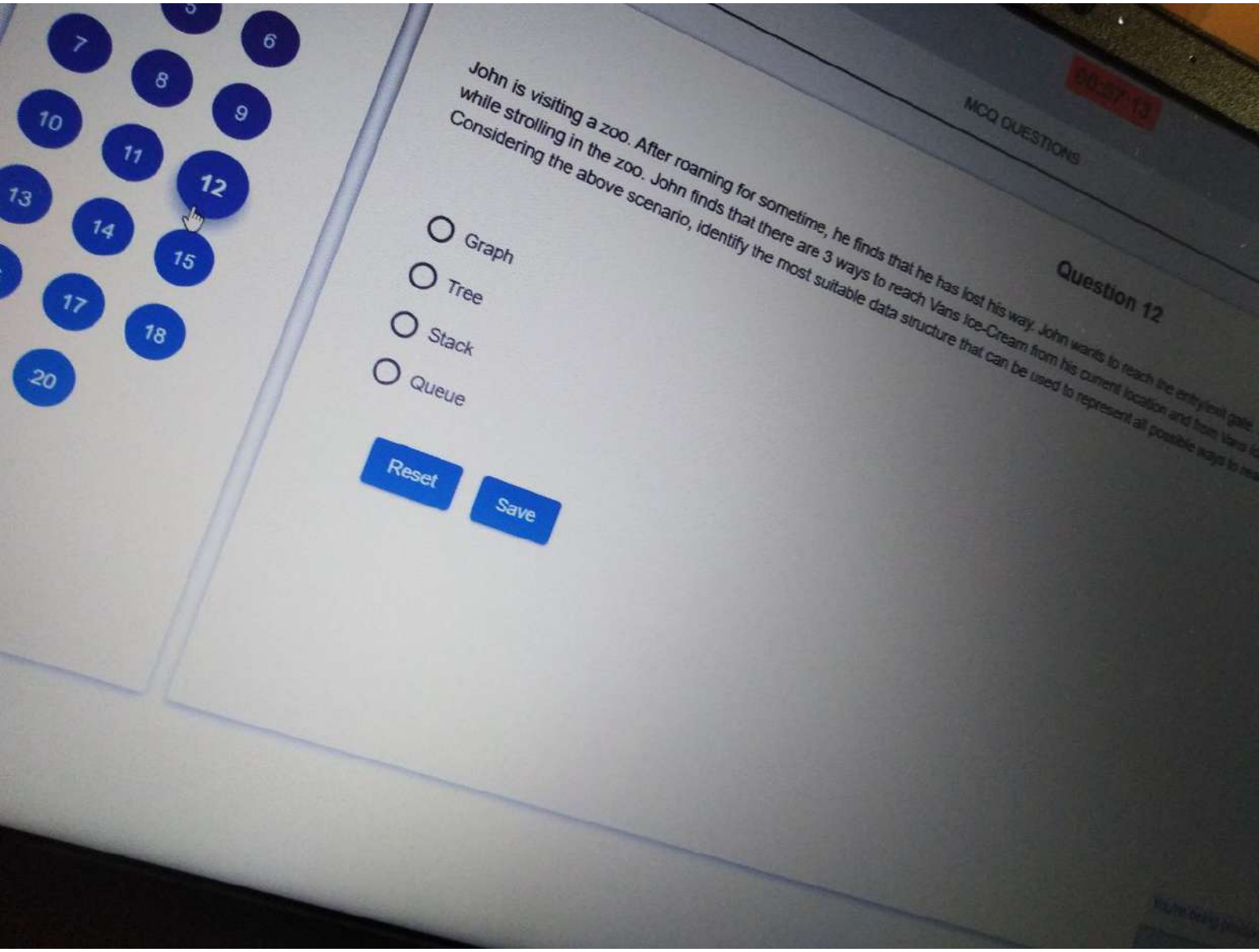

MCQ QUESTION8
From the below code snippet, identify how many reference variable(s) refer to the object created in Line 1 at the end of Line 2?
class Team:
counter=0
def __init__(self):
self.no_of_players=11
self.coach=None
cricket_team=Team()
football_team=Team()
hockey_team=football_team #Line1
football_team=cricket_team
cricket_team=Team()
cricket_team=hockey_team #Line2
Note: Line numbers are only for reference
3
2
0
1
Reset
Save


MCQ QUESTIONS
Question 18
Consider the below inputs:
input_linked_list (Head to Tail): 1 ->2 ->5 ->3
input_stack (Top to Bottom): 4, 2, 5, 10
def generate(input_linked_list, input_stack):
temp= input_linked_list.get_head()
#get_head() returns the head node
element=0
while(temp.get_next() is not None):
#get_next() returns the address of the next node
temp.set_data(temp.get_data()+temp.get_next().get_data()+element)
#get_data() returns the data stored in the node
if temp.get_data()%2!=0:
temp.set_data(temp.get_data()+input_stack.pop())
#set_data(data) updates the data stored in the node
else:
element=temp.get_data()
input_stack.push(element)
element=temp.get_next().get_data()
temp=temp.get_next()
temp.set_data(temp.get_data()+input_stack.pop())
What will be the content of input_linked_list from head to tail and input_stack from top to bottom after the execution of the function generate
Assumption: Stack and LinkedList classes, with the necessary methods, are available
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20


MCQ QUESTIONS
Question 18
Consider the below inputs:
input_linked_list (Head to Tail): 1 ->2 ->5 ->3
input_stack (Top to Bottom): 4, 2, 5, 10
def generate(input_linked_list, input_stack):
temp= input_linked_list.get_head()
#get_head() returns the head node
element=0
while(temp.get_next() is not None):
#get_next() returns the address of the next node
temp.set_data(temp.get_data()+temp.get_next().get_data()+element)
#get_data() returns the data stored in the node
if temp.get_data()%2!=0:
temp.set_data(temp.get_data()+input_stack.pop())
#set_data(data) updates the data stored in the node
else:
element=temp.get_data()
input_stack.push(element)
element=temp.get_next().get_data()
temp=temp.get_next()
temp.set_data(temp.get_data()+input_stack.pop())
What will be the content of input_linked_list from head to tail and input_stack from top to bottom after the execution of the function generate
Assumption: Stack and LinkedList classes, with the necessary methods, are available
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
20
Question 10
MCQ QUESTION
Consider the BUBBLE SORT algorithm given below which sorts the given input_list in ASCENDING order
def bubble_sort(input_list):
num = len(input_list)
for index1 in range(num):
for index2 in range(0, num-index1-1):
if input_list[index2] > input_list[index2+1] :
input_list[index2], input_list[index2+1] = input_list[index2+1], input_list[index2]
Consider the following input_list
input_list=[4, 10, 2, 7, 1, 8]
What will be the content of input_list after third pass, when the above bubble_sort function is invoked by passing input_list?
[2, 4, 1, 7, 8, 10]
[2, 1, 4, 7, 8, 10]
[2, 4, 7, 1, 8, 10]
[1, 2, 4, 7, 8, 10]
Reset
Save


5
6
7
8
9
10
11
12
13
14
15
17
18
20
MCQ QUESTIONS
Question 13
Consider the below Python code:
class Glove:
def __init__(self,color):
self.__color = color
def get_color(self):
return self.__color
def set_color(self, color):
self.__color = color
class Minion:
def __init__(self,glove):
self.__glove = glove
self.__color = "Yellow"
def get_glove(self):
return self.__glove
black_glove = Glove("Black")
red_glove = Glove("Red")
bob = Minion(black_glove)
black_glove.set_color(red_glove.get_color())
print(_____)#line1
What should be placed in blank at #Line1 to get the color of bob Minion's glove?
Note: Line number is for reference only.


7
6
8
10
9
11
12
13
14
15
17
18
20
MCQ QUESTIONS
Question 12
John is visiting a zoo. After roaming for sometime, he finds that he has lost his way. John wants to reach the entry/exit gate
while strolling in the zoo. John finds that there are 3 ways to reach Vans Ice-Cream from his current location and from Vans
Considering the above scenario, identify the most suitable data structure that can be used to represent all possible ways to
Graph
Tree
Stack
Queue
Reset
Save

MCQ QUESTIONS

00:57:23

Question 10

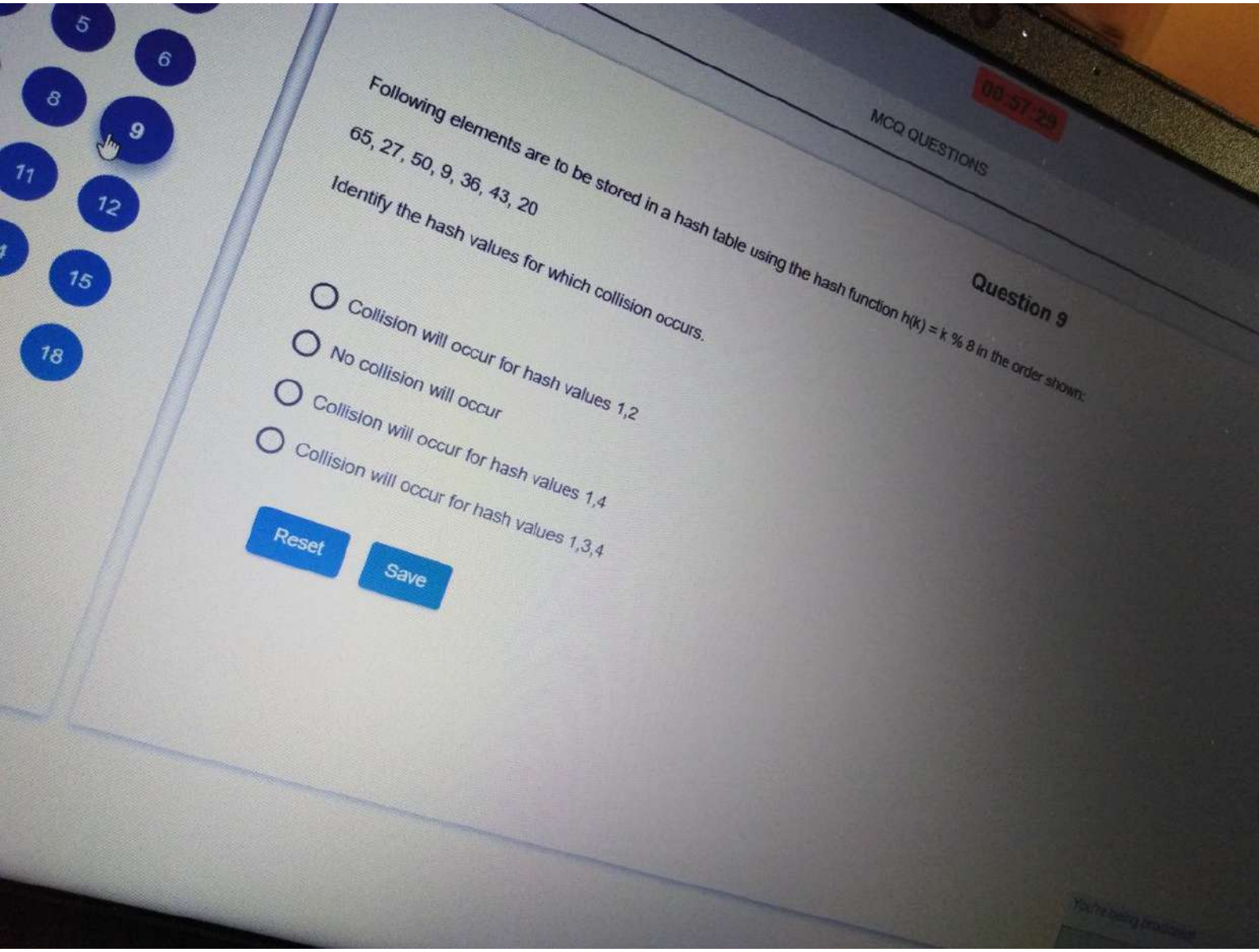
Consider the following Python function, the objective of the code is to record the details of service provider

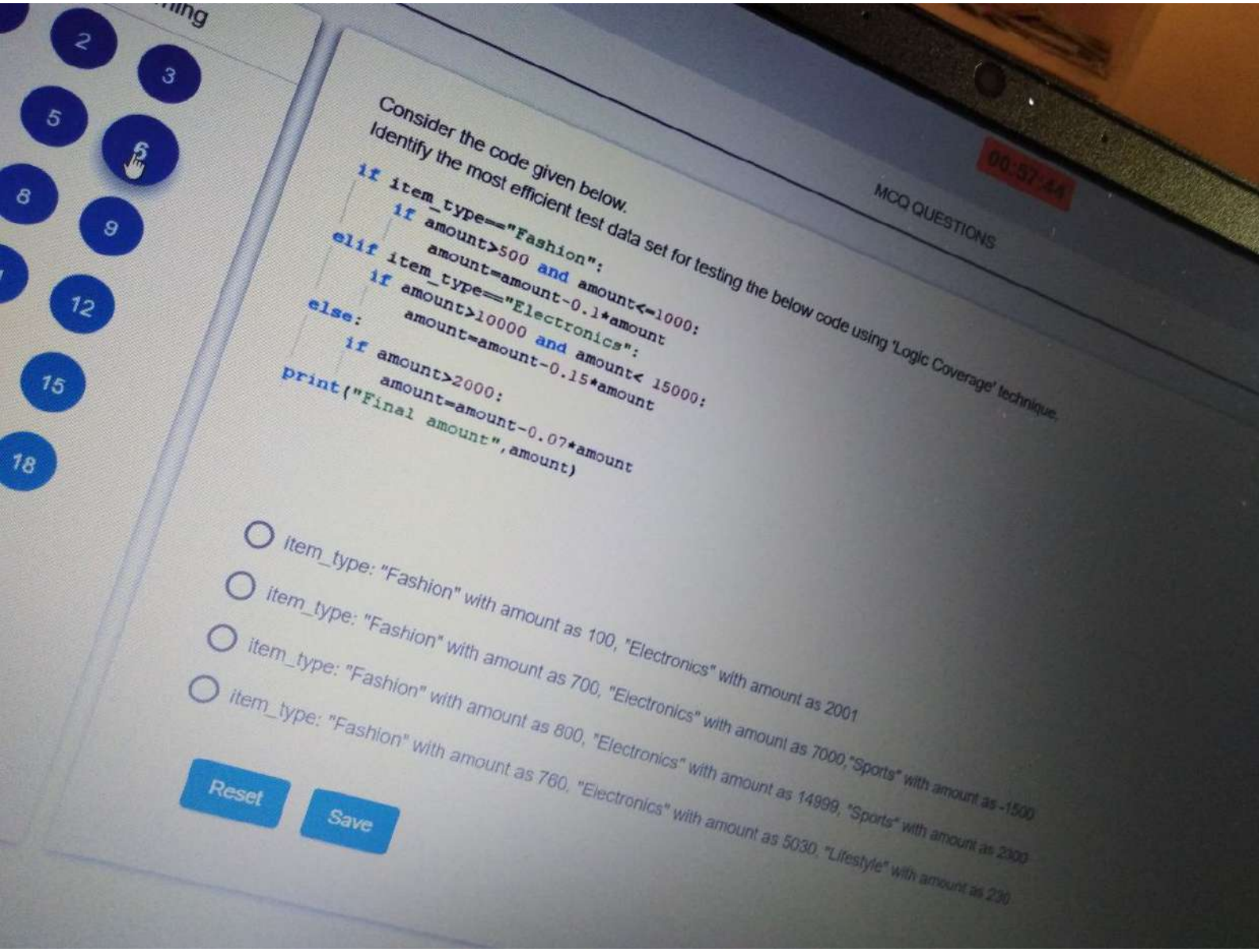
```
def service_provider_details(service_provider, tariff_plan, expiry_date, channel_pack,
                             only_calls, service_type, data_charge, unlimited_data):
    plan="Expiry date is "+expiry_date
    if(service_provider=="DTH"):
        plan+="with tariff plan "+ tariff_plan+ " and with"+channel_pack
    elif(service_provider=="MobileService"):
        plan+="with"+only_calls
        if(service_type=="PrepaidService"):
            plan+="Data at per MB charge"+data_charge
        elif(service_type=="PostpaidService"):
            plan+="with data offer"+unlimited_data
```

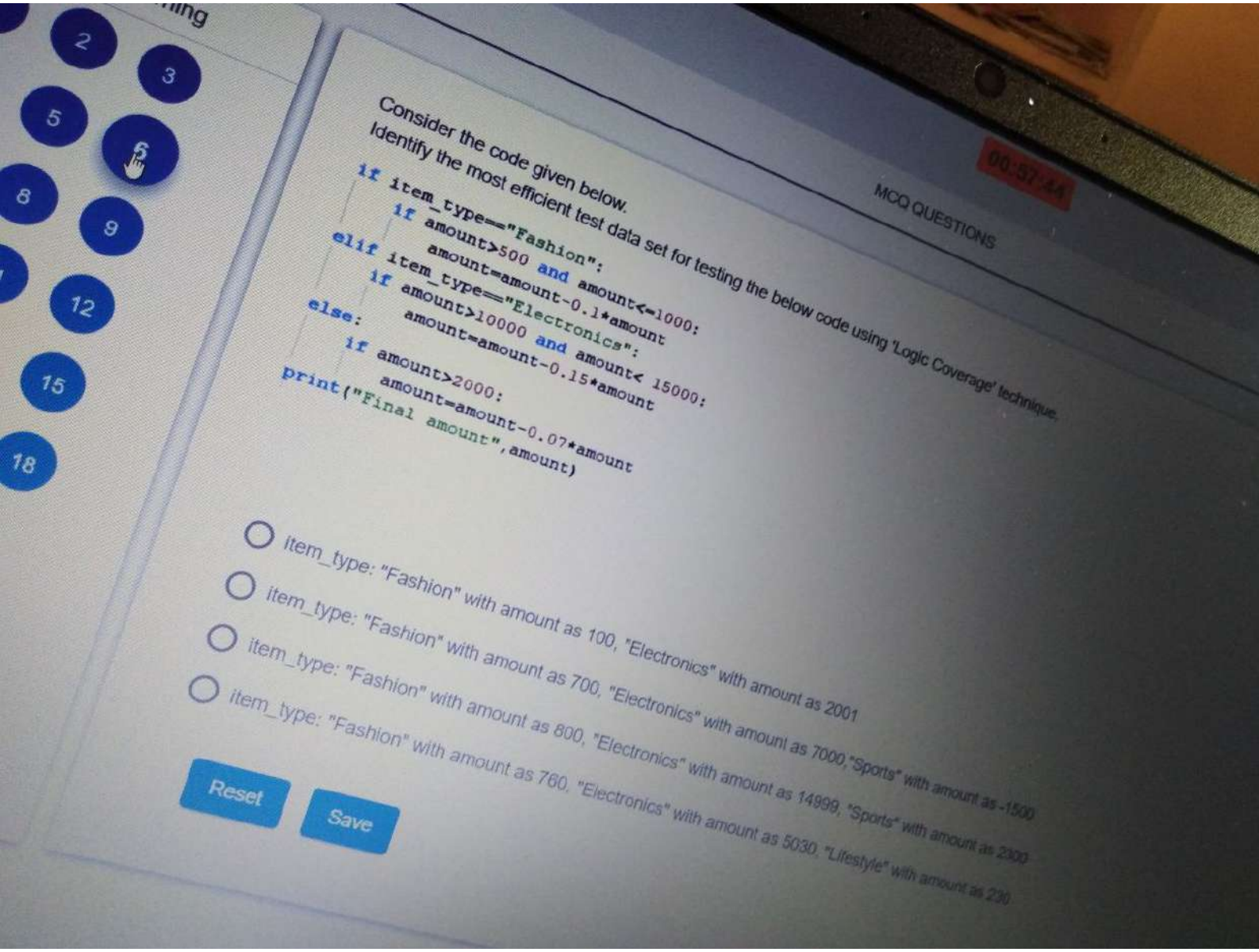
What will be the optimal class structure if this was to be rewritten in Object Oriented Programming?

- 4 classes: MobileService as the parent class. DTH, PrepaidService and PostpaidService as the child classes of MobileService
- 5 independent classes: ServiceProvider, DTH, MobileService, PrepaidService and PostpaidService
- 5 classes: ServiceProvider as the parent class. DTH and MobileService as the child classes of ServiceProvider. PrepaidService and PostpaidService as the child
- 4 classes: ServiceProvider as the parent class. DTH, PrepaidService and PostpaidService as the child classes of ServiceProvider

Reset Save


5
6
8
9
11
12
15
18
00:57:29
MCQ QUESTIONS
Question 9
Following elements are to be stored in a hash table using the hash function h(k) = k % 8 in the order shown:
65, 27, 50, 9, 36, 43, 20
Identify the hash values for which collision occurs.
Collision will occur for hash values 1,2
No collision will occur
Collision will occur for hash values 1,4
Collision will occur for hash values 1,3,4
Reset
Save


MCQ QUESTIONS
00:57:44
Consider the code given below.
Identify the most efficient test data set for testing the below code using 'Logic Coverage' technique.
if item_type=="Fashion":
if amount>500 and amount<=1000:
amount=amount-0.1*amount
elif item_type=="Electronics":
if amount>10000 and amount< 15000:
amount=amount-0.15*amount
else:
if amount>2000:
amount=amount-0.07*amount
print("Final amount",amount)
item_type: "Fashion" with amount as 100, "Electronics" with amount as 2001
item_type: "Fashion" with amount as 700, "Electronics" with amount as 7000,"Sports" with amount as -1500
item_type: "Fashion" with amount as 800, "Electronics" with amount as 14999, "Sports" with amount as 2300
item_type: "Fashion" with amount as 760, "Electronics" with amount as 5030, "Lifestyle" with amount as 230
Reset
Save


MCQ QUESTIONS
00:57:44
Consider the code given below.
Identify the most efficient test data set for testing the below code using 'Logic Coverage' technique.
if item_type=="Fashion":
if amount>500 and amount<=1000:
amount=amount-0.1*amount
elif item_type=="Electronics":
if amount>10000 and amount< 15000:
amount=amount-0.15*amount
else:
if amount>2000:
amount=amount-0.07*amount
print("Final amount",amount)
item_type: "Fashion" with amount as 100, "Electronics" with amount as 2001
item_type: "Fashion" with amount as 700, "Electronics" with amount as 7000,"Sports" with amount as -1500
item_type: "Fashion" with amount as 800, "Electronics" with amount as 14999, "Sports" with amount as 2300
item_type: "Fashion" with amount as 760, "Electronics" with amount as 5030, "Lifestyle" with amount as 230
Reset
Save

00:57:53

MCQ QUESTIONS

Question 4

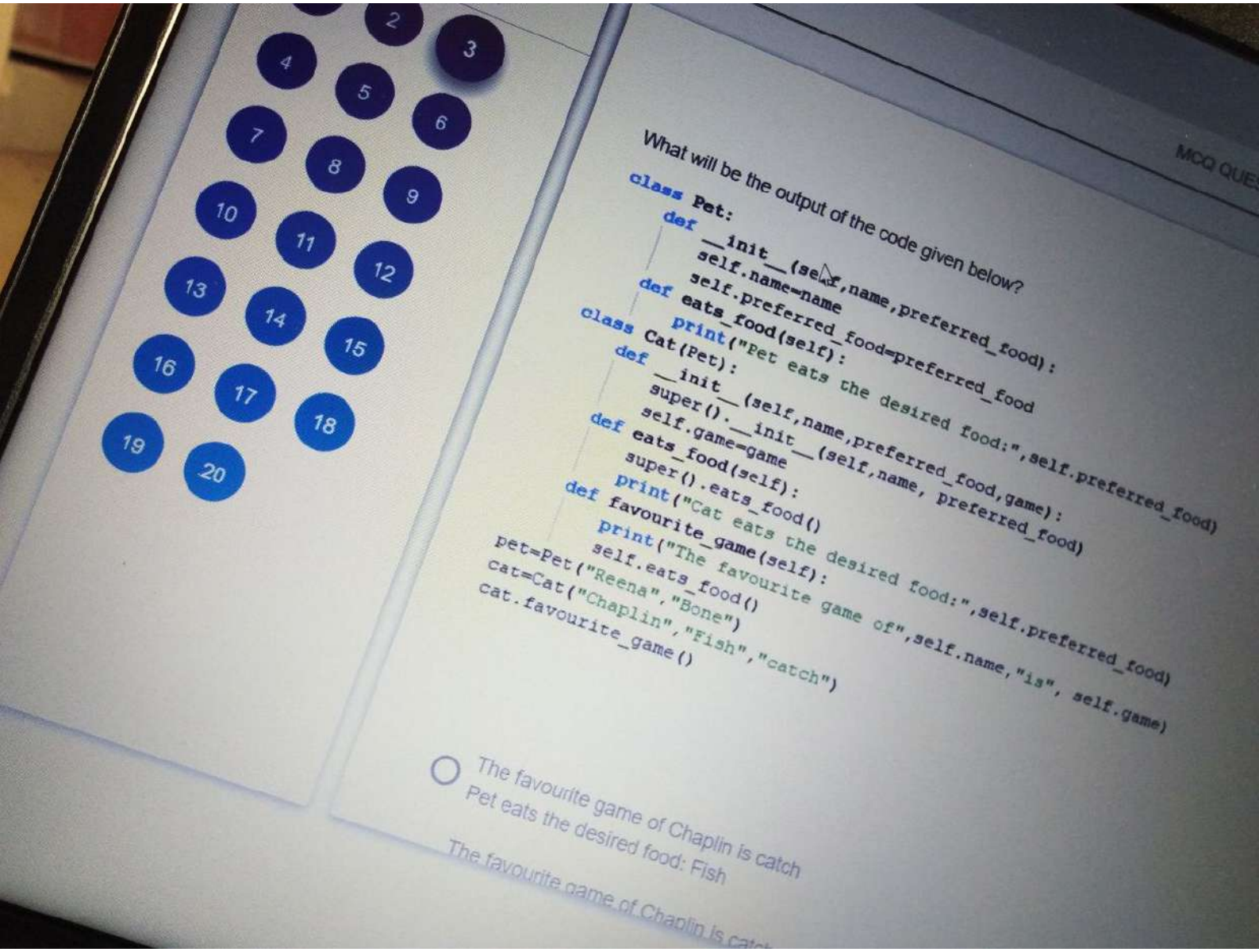
Consider the code given below:

```
def sample_func(var1,var2,var3=10,var4=3,*var5):
    var4 +=2
    var2 = var1*var4
    for val in var5:
        val = val*val
    print(var1,var2,var3,var4,var5)
sample_func(0,0,1,2,3,4,5)
```

Considering the function call and the function signature in the above code, which of the following values will be assigned to the formal argument var5 ?

- (1, 2, 3, 4, 5)
- (3, 4, 5)
- (4, 5)
- Error in function call

Reset Save


MCQ QUE
What will be the output of the code given below?
class Pet:
def __init__(self,name,preferred_food):
self.name=name
self.preferred_food=preferred_food
def eats_food(self):
print("Pet eats the desired food:",self.preferred_food)
class Cat(Pet):
def __init__(self,name,preferred_food,game):
super().__init__(self,name, preferred_food)
self.game=game
def eats_food(self):
super().eats_food()
print("Cat eats the desired food:",self.preferred_food)
def favourite_game(self):
print("The favourite game of",self.name,"is", self.game)
self.eats_food()
pet=Pet("Reena","Bone")
cat=Cat("Chaplin","Fish","catch")
cat.favourite_game()
The favourite game of Chaplin is catch
Pet eats the desired food: Fish
The favourite game of Chaplin is