**FULLSTACK ASSESSMENT - 2 HOURS**

| **SECTION TYPE** | **TOTAL MARKS AVAILABLE NOTES** |
| --- | --- |
| **Redux (pseudocode code/ reasoning explanation)** | 25 Multiple questions, all  comprising 25 total |
| **Algorithms 1 (Coding)** | 15 1 question only |
| **Algorithms 2 (Coding)** | 20 1 question only |
| **60 marks available total** | |

*Questions begin on the next page*

| **Redux** | **25 MARKS** |
| --- | --- |

This question involves using built-in React Hook for a simple attendance app.

*Here are notes to help: -*

*● useState() : This allows you to have states variables in functional components. It helps to set and retrieve the state.*

*● A reducer: This is a function that returns some state data, triggered by an action type. ● An action: This is dispatched by components and is represented as one object that*

*contains type property and sometimes payload property. It tells the reducer how to*

*change the state. Here is an example of the shape of an action -> { type: 'GREETINGS',*

*payload: ‘Hello’}*

*● Dispatch: this helps update the state by sending the type of action to the reducer*

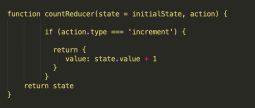
*function for it to perform its job. To invoke a dispatch function, you need to pass*

*action as an argument to the dispatch function, e.g. dispatch ({type: "SOMETHING",*

*payload: “SOMETHING” })}*

*\*\* Remember: Submit pseudocode or simply describe the solution.*

**Part 1 (5 marks):**

****

*Figure 1*

1. Can you provide a brief summary of what is happening in this function code?   
  
This code shows the reducer. If the action type that has been dispatched is “increment”, then it will increase the current state value by 1

2. Add one action that tells the reducer to reduce the state value by 1

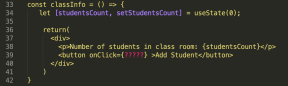
| if (action.type === 'decrement') {  return {  value: state.value - 1  } } |
| --- |

3. Add one action that tells the reducer to reset the state

| if (action.type === 'reset') {  return {  value: state.value = initialState *// potentially you could also set the value to 0,if this was the original value given*  } } |
| --- |

**Part 2 (10 marks):**

This section involves handling state locally.

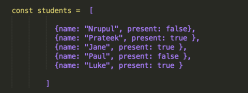


*In the code above the useState hook is used to set the state of a variable inside the component.*

1. Can you provide a brief summary on what is happening on line 34, 39?

On line 34 we’ve declared the useState hook called studentsCount. It is initialised with a value of 0. Then in the JSX underneath, the number of students currently counted will be displayed using this value, and the onClick button action should add one to the studentsCount each time it’s pressed.

2. When a user clicks on the “Add student” button update the state (studentsCount) to include only the total number of students who are present. Using the data provided below:



*Figure 3*

a. Write a *pseudocode* of how your function would look.   
  
 create function addStudents () = {

start for loop to iterate through the array {

if the value of present is true: {

setStudentsCount to studentsCount + 1

}

}

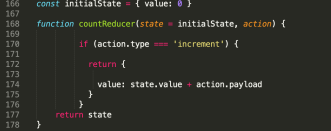
}

b. How do you ensure that the function is triggered when the button is clicked?   
  
 You’d update the onClick code to contain the function we’d just written, e.g. onClick={addStudents}

c. How will you update the state with the result of your function?   
  
 By using the setState part of the useState hook that’s already been declared on line 34. Something like this: setStudentsCount(studentsCount + 1)

**Part 3 (10 marks):**

Now let’s use dispatch to update the state on button click



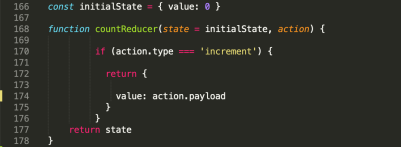
*Figure 4*

1. A change of code was made on line 174 (figure 4), can you briefly explain what that would do?   
  
This would update the state value, with the value passed by the payload. The payload itself is set elsewhere in the code. This will increment the current count by whatever value is passed from the payload, and this updated value is what will be returned.

2. Let’s say we don’t want to set the state locally anymore and want to use dispatch. How would you ensure that an “*increment*” action that also contains the result of the studentCount is dispatched on button click? According to your answer in part 2.2b what would need to be changed?  
  
We would need to include the dispatch action within our function. We’d have to include a line, like:  
dispatch({type: ‘increment’, payload: 1}). This could sit within the if statement portion and only run, when a student’s presence is true.

3. Which code do you think is best suited to ensure that the “increment” action updates the state with the *correct* total number of students who are present. *Is it Figure 4? Or*

*Figure 5? Explain the code difference and your reasoning*

**

*Figure 5***I think figure 4 would be correct**. The code in figure 4, adds the payload to the current state total, which should increase the count for each student who is added.  
  
It looks like the code in figure 5 just returns the payload, without adding it to the overall count - so in theory it would always return the same value instead of incrementing up.

| **Algorithms 1 (Coding)** | **15 MARKS**  **(1 question)** |
| --- | --- |

Write an algorithm that returns true if the given string is a palindrome. Otherwise, return false.

*Note:* A String is said to be a palindrome if the string is spelled the same way forward and backwards.

For example, some sample input and outputs would be:

|  |
| --- |
| **Sample Input 1** |
| **Sample Input 2** |
| **Sample Input 3** |
| **Sample Input 4** |

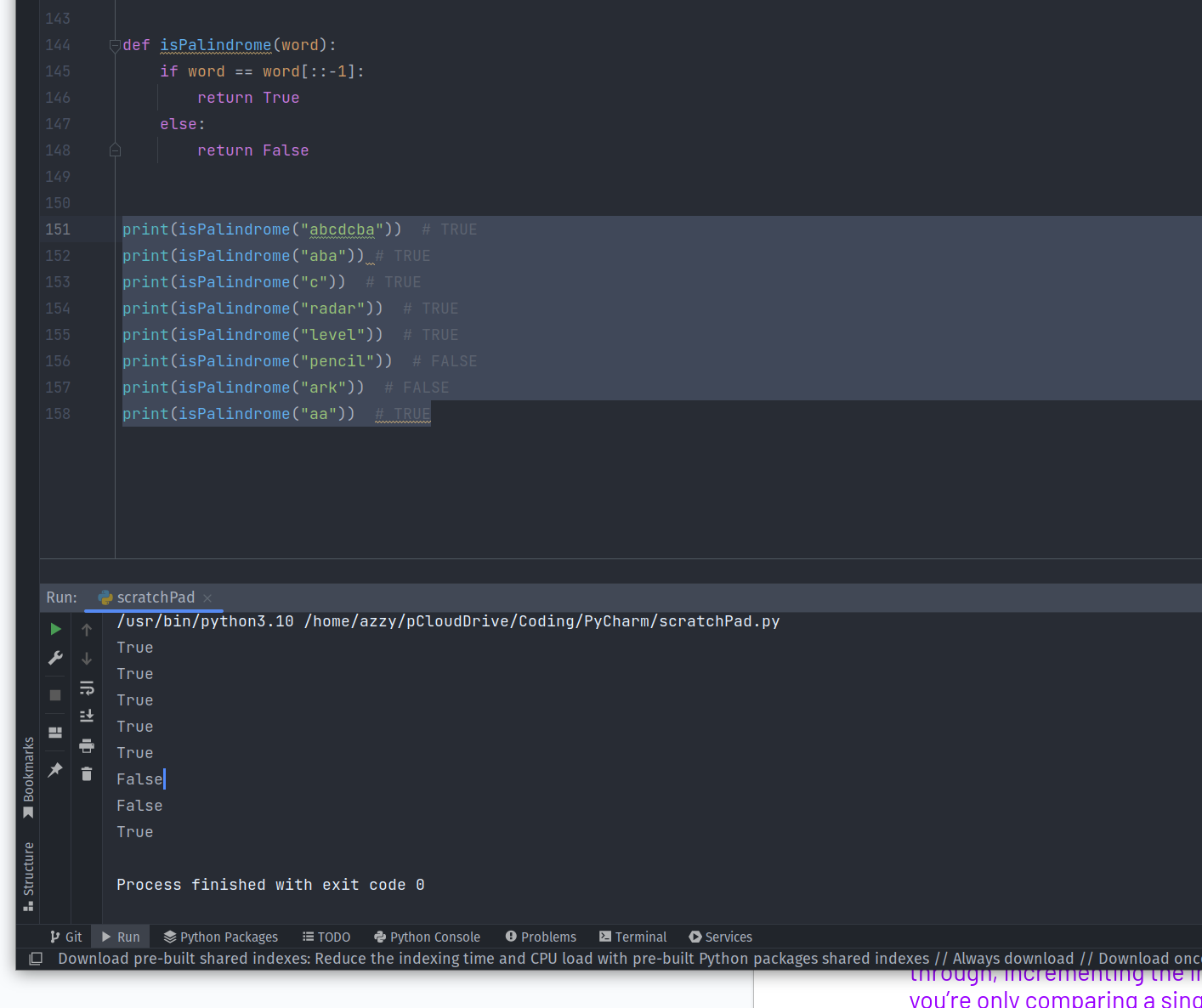
**stringA value Output value** radar True level True Pencil False a True

*In your answer, please discuss your solution - what is its Big O Time & Space complexity? Why have you chosen this approach? Could there be a more efficient way (and if so, how)?*

*If you are short on time, you can also submit pseudocode or simply describe what solution you’d write in code (just describe what you have in your mind) - this cannot attain full marks, but it is still a perfectly acceptable answer and can get partial marks.*

*In essence, just submit what you have even if you don’t know the answer!   
  
Code:*

| *def isPalindrome(word):  if word == word[::-1]:  return True  else:  return False  print(isPalindrome("abcdcba")) # TRUE print(isPalindrome("aba")) # TRUE print(isPalindrome("c")) # TRUE print(isPalindrome("radar")) # TRUE print(isPalindrome("level")) # TRUE print(isPalindrome("pencil")) # FALSE print(isPalindrome("ark")) # FALSE print(isPalindrome("aa")) # TRUE* |
| --- |



This code is simple, and easily memorised (which is why I chose it as my solution, as I can’t remember how to do more elegant code from memory). It takes the word, reverses it, and sees if the two strings match, returning true or false in either case.

I think this is probably O(n) in terms of the time complexity - as we aren’t looping, or doing anything which would make it more complex. If the string gets longer, the time taken to process it, should increase at the same rate.

I know there’s more efficient ways of solving this which use less memory. This solution isn’t ideal because it creates a second strong and so it takes both more space and time. There’s more elegant solutions which involve looking at just the first and last character, comparing them, and iterating through, incrementing the index on the left and right sides both times. This is more efficient, because you’re only comparing a single character, and stopping straight away if no match is found. And you’re only comparing a character at a time, compared to my clunky solution. Because you can eliminate characters, that could potentially be logarithmic? But it would certainly use less space than my solution, even it was O(n) in time.

| **Algorithms 2 (Coding)** | **20 MARKS**  **(1 question)** |
| --- | --- |

Write a function that takes in an unsorted array of any size. These elements are in the range of 1 to n. In the input array one number is missing. Your function should return the missing number.

If the input array contains a negative number or non-numeric value then return an error with the correct error message.

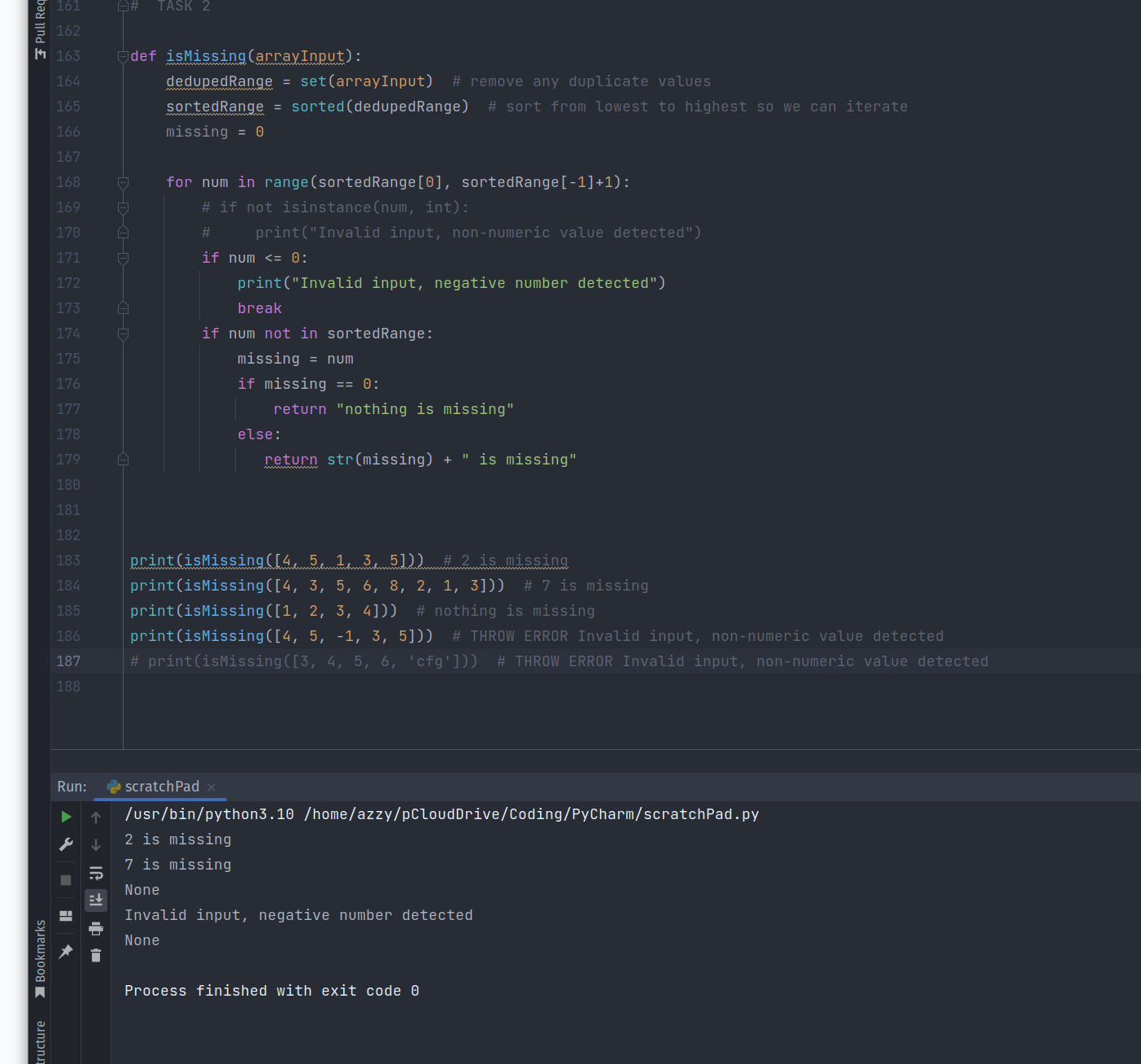
For example, some sample input and outputs would be:

|  | **Array input Output** |
| --- | --- |
| **Sample Input 1** | [4,5,1,3, 5] Missing = 2 |
| **Sample Input 2** | [4, 3,5, 6, 8, 2, 1, 3] Missing = 7, |
| **Sample Input 3** | [1,2,3,4] “Nothing is missing” |
| **Sample Input 4** | [4,5, -1,3, 5] “Invalid input, negative number detected” |
| **Sample Input 5** | [ 3, 4, 5, 6, 'cfg' ] “Invalid input, non-numeric value detected” |

*In your answer, please discuss your solution - what is its Big O Time & Space complexity? Why have you chosen this approach? Could there be a more efficient way (and if so, how)?*

*If you are short on time, you can also submit pseudocode or simply describe what solution you’d write in code (just describe what you have in your mind) - this cannot attain full marks, but it is still a perfectly acceptable answer and can get partial marks.*

*In essence, just submit what you have even if you don’t know the answer!*

**

| *def isMissing(arrayInput):  dedupedRange = set(arrayInput) # remove any duplicate values  sortedRange = sorted(dedupedRange) # sort from lowest to highest so we can iterate  missing = 0   for num in range(sortedRange[0], sortedRange[-1]+1):  # if not isinstance(num, int):  # print("Invalid input, non-numeric value detected")  if num <= 0:  print("Invalid input, negative number detected")  break  if num not in sortedRange:  missing = num  if missing == 0:  return "nothing is missing"  else:  return str(missing) + " is missing"    print(isMissing([4, 5, 1, 3, 5])) # 2 is missing print(isMissing([4, 3, 5, 6, 8, 2, 1, 3])) # 7 is missing print(isMissing([1, 2, 3, 4])) # nothing is missing print(isMissing([4, 5, -1, 3, 5])) # THROW ERROR Invalid input, non-numeric value detected # print(isMissing([3, 4, 5, 6, 'cfg'])) # THROW ERROR Invalid input, non-numeric value detected* |
| --- |

I ran out of time to finish my code and only some of my cases work, but you can see I was building in functionality to check if numbers were not integers (as in the bottom test case), and functionality to check whether a number was less than 0 and record an error in that case too. Almost there but just ran out of time.

I wanted to minimise the number of for loops to try and keep my code efficient, as introducing more loops would turn this into quadratic in terms of the time it takes. I believe this code as it is, would be O(n)?

I was trying to iterate through the array, after having removed any duplicates, and look for the total length of the array, spotting the missing number. It works for this, but not for the edge cases of negatives, non-integers, etc.

Once working, I’d like to re-write this as a list comprehension if possible, to make it cleaner and more efficient.