

Unit 1- Introduction

Q1

I have a question about the algorithms. What happens if the N value used goes over the number of elements in the array? Do we just add on from the first number again?

Example:

s = (1,2,3,4,5), N = 7

```
array_Sum (s, n) {  
    sum = 0  
    for (i = 0 to (n-1) ) {  
        sum = sum +s[i]  
    }  
    return sum  
}
```

>>>> Answer: No. It all depends on the algorithm called. For your example above, for array_Sum ([1,2,3,4,5], 7), when the for loop is executed, when i reaches 5, the statement “sum = sum +s[i]” will want to refer to the element indexed by 5 in the input array. But there is no such element as the array only has 5 elements. So, the algorithm will be abnormally terminated. And, it will not return any value for sum. Hence, sum is undefined.

Also, if there are Mutiple returns in the algorithm, it only returns 1 answer? Also, if a return statement ends the algorithm, that means anything below it is not used?

>>>> Answer: Yes, you are right.

Lastly, are there any other IT techniques to deal with Wicked problems, because ones in the slides seem to be more on mapping.

>>>> Answer: The techniques in our notes are some IT techniques that help to deal with wicked problems. Certainly, more tools can be developed or invented but these are beyond our scope

Q2

I am a student in one of your class and I would like to ask what does the “[]” symbol means in the following example below.

Example: find the largest element in an array

Input array A of n integers

Output largest element in A

```
arrayMax(A, n) {  
    Max = A[0]  
    for (i = 1 to n - 1)  
        if A[i] > Max then  
            Max = A[i]  
    return Max  
}
```

	0	1	2	3	4	
A	9	10	25	34	6	Max: 34

>>>> Answer: In the above algorithm, $A[1]$ refers to the array element indexed by 1 – that is the second element. Note that $A[1] = 10$. $A[0]$ refers to the array element indexed by 0 – that is the first element. Note that $A[0] = 9$. $A[4]$ refers to the array element indexed by 4 – that is the fifth element. Note that $A[4] = 6$, etc...

Q3

On testReturn below, what is the value returned when $y=4$?

And, what is the value returned when $y=10$?

What is/are the value/values returned by the following algorithm when y is 4?
What is/are the value/values returned by the following algorithm when y is 4?

```
testReturn(y) {  
    if (y > 5)  
        return y  
    if y = 10  
        return (y+10)  
    return (y+20)  
    return (y+30)  
    return (y+40)  
}
```

Select one:

- ☐ a. 4
- ☐ b. 34
- ☐ c. 24
- ☒ d. 24, 34 and 44 ✖
- ☐ e. 14
- ☐ f. 44

The correct answer is: 24

>>> Answers: Executing this algo with y is 4, first, as it does not satisfy the conditions in the two ifs, hence, the actions under the two if will not be executed.

Consequently, it will execute the statement "return (y+20)". It returns $(4 + 20) = 24$ **and exit and stop the execution.** No further statements will be executed.

Hence, all the return statements after "return (y+20)" will not be executed.

Important Note: When a return statement is executed, the execution STOP and EXIT FROM THE ALGO and NO FURTHER STATEMENT WILL BE EXECUTED.

>>> Answers: When y is 10, it satisfies the condition of the second, hence, it executes its action the return statement and return 20 and exit from the algo.

When a return statement is executed, the execution STOP and EXIT FROM THE ALGO and NO FURTHER STATEMENT WILL BE EXECUTED.

Q4

For the following algo:

```
The algorithm array_mmm1(A, n) is shown below. What is the value returned by array_mmm1([1, 3, 4, 5], 4)?

array_mmm1(s, n) {
    large = s[0]
    i = 1
    while (i < n) {
        if (s[i] ≥ large) {
            large = s[i]
            return large
        }
        i = i + 1
    }
}

Select one:
☒ a. 5 ✖
☐ b. 4
☐ c. 3
☐ d. 1
☐ e. 0
```

For this algo, as the action of the if statement includes return statement. In the execution of array_mmm1([1, 3, 4, 5], 4), the return statement is executed in the first iteration. So, it return s[1] = 3 and stop the exec.

Q5

For the following algo:

```
array_mmm2(s, n) {
    large = s[0]
    i = 1
    while (i < n) {
        if (s[i] ≥ large)
```

```

        large = s[i]
        i = i + 1
    }
    return large
}

```

>>>> Answers: In this case, as the action for while loop does not include any return statement, hence, the execution of array_mmm2([1, 3, 4, 5], 4) will execute the while loop 3 times, on the last time, it assigns large = s[3], hence, return 5,

Q6

The algorithm findTarget(A, n) is shown in Figure 4. What is the value returned by findTarget([3, 15, 7, 2], 4)?

```

findTarget(s, n) {
    EleFound = s[0]
    for (i = (n-1) downto 0 ) {
        if (s[i] > EleFound)
            EleFound = s[i]
        return EleFound
    }
}

```

>>>> Answer: In the execution of findTarget([3, 15, 7, 2], 4), initial set EleFound = s[0] = 3, and during the first iteration of the for loop i = 3, so the if condition is not satisfied, Elefound remains at 3. And, as return statement is in the for loop, next, it will execute the return statement and return 3. Then, complete the execution.

Q7

Sorry to disturb you, regarding algorithm, as there are different ways to write algorithm, I wonder if my following algorithm is correct if I choose to write it this way. It is to add up all the values in an array. Thanks a lot for your help.

```

sumArray(s,n){
    i = n-1
    sum = 0
    for(s[i]down to 0)

```

```

    {
        sum=(sum+s[i])
        return sum
    }
    return sum
}

```

>>>> Answer: No. There are two major mistakes: (1) when the for loop is first executed, your for loop starts from s[i] which is the value of the last element in the array s. This is wrong. Your for loop should be "for (i = (n-1) down to 0)", Then, each execution of the for loop will add an element in the array s from the last element until the first element. (2) The first return statement in your for loop should be removed. Otherwise, the sum will be returned as the value of the first element in the array s. This is because once a return statement is executed, the execution will be completed, no further statement will be executed.

Q8

I have a question regarding pseudocode. What is the difference between the 2 code where the } at the end is placed differently.

1.

```

Find3(s, n) {
    EleFound = s[0]
    for (i = (n-1) downto 0 ) {
        if (s[i] < EleFound)
            EleFound = i
    }
    return EleFound
}

```

2.

```

Find3(s, n) {
    EleFound = s[0]
    for (i = (n-1) downto 0 ) {
        if (s[i] < EleFound)
            EleFound = i
    }
    return EleFound
}

```

>>>> Answer: Due to the different placing of }, there is a lot of difference between these two pieces of algos. The first algo will check all the elements in the array according to the if statement by iterating the actions in the for loop (n-1) times. After the for loop execution is completed, it will return the value of EleFound. The second one will just return (n-1) if cond is satisfied, otherwise, it will return s[0]. The for loop is only iterated 1 time. This is because once a return statement is executed, the execution will be completed, no further statement will be executed.

Q9

I'm a fulltime student in your CSIT113 module. It's my first time doing pseudocoding and I have some questions regarding for loop in the quiz.

The algorithm AddMul6(A, n) is shown below. What is the value returned by AddMul6([7, 5, 7, 9], 4)?

```
AddMul6(A, n) {  
  sum = A[0]  
  for i = (1 to (n-1)) {  
    sum = sum + i  
  }  
  return sum  
}
```

>>>> Answer: For AddMul6([7, 5, 7, 9], 4), initially sum set to A[0] = 7. Then it executes the for loop start from i = 1. So in the first iteration of for loop, it adds 1 to sum, so sum is set to 8. Then it returns 8. And, the algo stops (execution is completed). No further execution of any statement in the algo. This is because once a return statement is executed, the execution will be completed, no further statement will be executed.

Q10

For the i = n-1, is it possible to state earlier instead of inside the for loop? For easier reading purposes? Like this

```
sumArray(s,n){  
  i = n-1  
  sum = 0  
  for([i]down to 0)  
  {  
    sum=(sum+s[i])  
  }  
  return sum  
}
```

>>> Answer: No. The for loop is a control structure. You have to follow the syntax below:

```
for (var = init downto limit)  
  action
```

So, your for loop is written wrongly. It should be as follows:

```
sumArray(s,n){
  i = n-1
  sum = 0
  for (i = (n-1)]down to 0)
  {
    sum=(sum+s[i])
  }
  return sum
}
```

Q11

For this question , can i check why the answer is 3 (9) , when the loop should continue till 1(3) since it is the smallest number.

Question 5
Incorrect
Mark 0.00 out of 1.00
Flag question

The algorithm Find3(A, n) is shown below. What is the value returned by Find3([15, 3, 7, 9], 4)?

```
Find3(s, n) {
  EleFound = s[0]
  for (i = (n-1) downto 0) {
    if (s[i] < EleFound)
      EleFound = i
  }
  return EleFound
}
```

Select one:

- ☐ a. 3
- ☐ b. 7
- ☐ c. 15
- ☐ d. 9
- ☒ e. 1

The correct answer is: 3

>>>> Answer: You must execute the algo Find3 statement by statement until its completion (stop and exit) to find the answer. For the given input $s = [15, 3, 7, 9]$, $n = 4$. The execution is as follows:

First, $\text{EleFound} = s[0] = 15$.

Next, the for loop is iterated from $i = (n-1)$ downto 0 that is $i = (4-1) = 3$ down to 0. In the first iteration $i=3$, EleFound is set to $i = 3$, as $s[3]=9 < 15$ it satisfies the if condition in the for loop.

In the 2nd iteration $i=2$, EleFound remains unchanged as 3 as it does not satisfy the if condition ($s[2] = 7$ is not less than 3) in the for loop.

In the 3rd iteration $i=1$, EleFound remains unchanged as 3 as it does not satisfy the if condition ($s[1] = 3$ is not less than 3) in the for loop.

In the last iteration $i=0$, Elefound remains unchanged as 3 as it does not satisfy the if condition ($s[0] = 15$ is not less than 3) in the for loop.

Now, for loop execution finished.

Then, it executes the statement after for loop, and return Elefound = 3.

Q12

May i know why the answer is not 15?

The screenshot shows a Moodle quiz review page for a user named Shun Hong Fergus Tay. The question is titled "Question 6" and is marked "Incorrect" with a score of "Mark 0.00 out of 1.00". The question text is: "The algorithm Find2(A, n) is shown below. What is the value returned by Find2([3, 15, 7, 9], 4)?" The algorithm code is as follows:

```
Find2(s, n) {
  EleFound = s[0]
  for (i = (n-1) downto 0) {
    if (s[i] > EleFound)
      EleFound = i
  }
  return EleFound
}
```

The user selected option "d. 15", which is marked incorrect with a red 'X'. The correct answer is "c. 0". A yellow box at the bottom of the question area states: "The correct answer is: 0". The left sidebar shows the course "CSIT113_SP121" and various navigation links. The bottom of the page shows a Windows taskbar with the time 5:32 pm on 2/1/2021.

>>> Answer: You must execute the algo Find2 statement by statement until its completion (stop and exit) to find the answer. For the given input $s = [3, 15, 7, 9]$, $n = 4$. The execution is as follows:

First, Elefound = $s[0] = 3$.

Next, the for loop is iterated from $i = (n-1)$ downto 0 that is $i = (4-1) = 3$ down to 0

In the first iteration $i=3$, Elefound = $i = 3$ as it satisfies the if condition in the for loop.

In the 2nd iteration $i=2$, Elefound = $i = 2$ as it satisfies the if condition in the for loop.

In the 3rd iteration $i=1$, Elefound = $i = 1$ as it satisfies the if condition in the for loop.

In the last iteration $i=0$, Elefound = $i = 0$ as it satisfies the if condition in the for loop.

Now, for loop execution finished.

Then, it executes the statement after for loop, and return 0.

Q13

Previously you assisted me on this question below.

What is the value returned by AddMul6([7, 5, 7, 9], 4)?

```
AddMul6(A, n) {  
  sum = A[0]  
  for i = (1 to (n-1)) {  
    sum = sum + i  
  }  
  return sum  
}
```

You said after the first iteration (first iteration of for loop, it adds 1 to sum, so sum is set to 8. Then it returns 8)

the algo stops because of the return statement (return sum). This is where I want to ask a question here. if there's no return statement and i move on to the 2nd iteration, the sum is 8. Does that mean that i = 2 ? so for the sum do i add the array value or the index? I'm pretty confuse at this point because I assume that the variable i is like a counter moving from point 1 to point 2 etc.

Also wanted to add another question here about the return statement where the algo stops. Base on that theory that if there's a return statement the algo stops, does it apply to the question below as well (meaning after the first iteration it stops)? Also you mention the braces matter as well.

I notice the above question has 2 sets of braces while the question below has 1. Does that explain about the way the return statement is being executed?

Not to mention this question has 5 elements while being given only 4 arrays index from 0-3. Does that mean it can only run 3 iterations? I tried solving it as per normal like this initial sum A[0] = 3, first iteration adds 1 to sum, so sum is set to 4. at this point i'm not sure if the algo stops because of the return statement there or should i continue.

what is the value returned by AddMul5([3, 1, 2, 4], 5)?

```
AddMul5(A, n) {  
  sum = A[0]  
  for i = (1 to (n-1))  
  sum = sum + i  
  return sum  
}
```

>>> Answer: My answer to each sub-question is listed below:

You said after the first iteration (first iteration of for loop, it adds 1 to sum, so sum is set to 8. Then it returns 8)

the algo stops because of the return statement (return sum). This is where I want to ask a question here. if there's no return statement and i move on to the 2nd iteration, the sum is

8. Does that mean that $i = 2$? so for the sum do i add the array value or the index? I'm pretty confuse at this point because I assume that the variable i is like a counter moving from point 1 to point 2 etc.

>>> Answer: this algo will add the index not the array element as the action under the for loop is "sum = sum + i". If you want to add the array elements, this statement should be changed to "sum = sum + s[i]". What values will be added is all depend on how the statement is written.

Also wanted to add another question here about the return statement where the algo stops. Base on that theory that if there's a return statement the algo stops, does it apply to the question below as well (meaning after the first iteration it stops)?

>>> Answer: yes certainly. The outcome of executing a statement depends on the meaning of the statement.

Also you mention the braces matter as well.

I notice the above question has 2 sets of braces while the question below has 1. Does that explain about the way the return statement is being executed?

>>>Answer: yes.

Not to mention the quiz question below has 5 elements while being given only 4 arrays index from 0-3. Does that mean it can only run 3 iterations? I tried solving it as per normal like this

initial sum $A[0] = 3$, first iteration adds 1 to sum, so sum is set to 4. at this point i'm not sure if the algo stops because of the return statement there or should i continue.

>>> Answer: No, the for loop in the quiz question below will be iterated for $i = 1$ to $(5-1) = 4$ regardless the no of elements in the array.

In summary, how a statement is executed should be totally based on the meaning of the statement and where it is placed in the algo.

Q14

I have a few questions for you regarding the quiz.

The algorithm findTarget(A, n) is shown in Figure 4. What is the value returned by findTarget([3, 15, 7, 2], 4)?

```
findTarget(s, n) {  
    EleFound = s[0]  
    for (i = (n-1) downto 0) {  
        if (s[i] > EleFound)  
            EleFound = s[i]  
    }  
    return EleFound  
}
```

Figure 4. A simple algorithm

Does this mean that the program will return the element of s[i] or the index of s[i]

>>> Answer: The value returned by return statement is fully based on the variable followed the word “return”. Since in the above algo, this variable is “EleFound”, Therefore, it returns the value of EleFound when the return statement is executed. Note that in the above algo, EleFound is assigned set to s[0] initially and updated to s[i] in the for loop’s action. Hence, the value returned from EleFound is an element in the array, nothing to do with the indexes.

Q15

Hello Sir,

I understand that when I put braces at different places, it will make the algo different.

What I am not sure is when there is no curly brackets ‘{ }’ after the for loop, how should I interpret the question?

After looking at lecture slides, I would say the example algorithm is the same as the (1) algorithm but not (2) algorithm. (shown below)

Can I get a confirmation from you?

The algorithm AddMul5(A, n) is shown below, what is the value returned by AddMul5([3, 1, 2, 4], 5)?

```
AddMul5(A, n) {  
    sum = A[0]  
    for (i = 1 to (n-1))  
        sum = sum + i  
    return sum  
}
```

```
}
```

The following are just the same algorithm with additional curly brackets.

I have added a pair of curly brackets to the following (1) and (2) algorithm

So, is the above algorithm the same as:

(1)

```
AddMul5(A, n) {  
  sum = A[0]  
  for (i = 1 to (n-1)) {  
    sum = sum + i  
  }  
  return sum  
}
```

OR

(2)

```
AddMul5(A, n) {  
  sum = A[0]  
  for (i = 1 to (n-1)) {  
    sum = sum + i  
  }  
  return sum  
}
```

>>> Answer: If no braces after the for loop, while loop or if branch, it is interpreted as the action for the loop, while loop or if branch respectively has only ONE STATEMENT.

For example, the for loop in your question has only ONE statement `sum = sum + i`.

Yes, the algorithm extracted from Quiz is the same as the (1) algorithm but not (2) algorithm. (shown below)

Q16

Thank you for responding to my questions and I've understood explanations 1 and 3 however for 2 may I know what you mean by the comparison is only based on the last element of the array as the

algorithm mentioned if $A[\text{loopcount}] \neq B[\text{loopcount}]$, would this statement not mean that the loopcount for both A and B array are being compared?

The algorithm below is to determine if two given arrays A and B both with n integers have identical values. It returns a Boolean value "True" if the two arrays have identical values and "False" if otherwise. Two arrays A and B are considered having identical values if every element of array A is equal to its corresponding element in array B (A[0] is equal to B[0], A[1] is equal to B[1], etc.). Any problem with this algorithm?

```
CompareArray(A, B, n) {
    loopCount = 0
    Result = True
    while ((loopCount < n) {
        if A[loopCount] != B[loopCount]
            Result = False
        else
            Result = True
        LoopCount = LoopCount + 1
    }
    return Result
}
```

>>> Answer: You cannot just base on ONE statement to tell result return be an algorithm. To get the return value, you need to execute the statement by statement. from begin to stop. Please try to find return value for CompareArray([3, 4, 7], [999, 888, 7], 3), CompareArray([3, 4, 7], [66666, 00000, 7], 3), etc. In summary, in this algo, Result set by comparing any array element earlier always overridden by the checking of the checking of last element. Hence, the value of Result is solely depended on the checking of last element.

Q17

Is the symbol \neg representing not(\sim)?

>>>> Answer: Yes

Q18

Question 1

Incorrect

Mark 0.00 out of 1.00

Flag question

The algorithm AddMul5(A, n) is shown below, what is the value returned by AddMul5([3, 1, 2, 4], 5)?

```
AddMul5(A, n) {
    sum = 0
    for (i = 1 to (n-1))
        sum = sum + i
    return sum
}
```

Select one:

☐ a. 9

☐ b. 21

☐ c. 13

☐ d. 15

☒ e. 6

The correct answer is: 13

Why the correct answer is 13 for the following q?

>>>> Answer:

In execution AddMul5([3,1, 2, 4], 5):

First, sum = A[0] = 3

Next iterate the "for loop" for i = 1 to (5-1) = 4 as follows (note that this loop has only one action "sum = sum +i" as there is braces attached):

i = 1, sum = sum +1 = 3 +1 =4

i = 2, sum = sum + 2= 4 +2 = 6

i = 3, sum = sum +3 = 6 +3 = 9

i = 4, sum = sum + 4 = 9 +4 = 13

Then the "for loop" execution is completed and the return statement is executed to return 13

Q19

Won't this whole sequence work if you were to remove the Else statement?

```
CompareArray(A, B, n) {  
    loopCount= 0  
    Result =True  
    while ((loopCount < n) {  
        if A[loopCount] != B[loopCount]  
            Result = False  
        else  
            Result = True  
        LoopCount = LoopCount + 1  
    }  
    return Result  
}
```

>>>> Answer: Yes. Great. But, less efficient.

Q20

Is it necessary for algorithm to be properly indented? There are a few quiz examples posted that does not have proper indentation or can i assume that this is part of the test to test whether we can read properly without indentations?

>>> Answer: indentation is just to prove readability for humans. It does not add additional meaning. Hence, indentation does not change the meaning of pseudocode.

Q21

Question 3

Correct

Mark 1.00 out of 1.00

Flag question

The algorithm CompareSequence(A, B, n) is given below. What is the value returned by CompareSequence([4, 3, 7], [1, 3, 8], 2)?

```
CompareArray(A, B, n) {  
    loopCount = 0  
    Result = "True"  
    while ((loopCount < n) {  
        if A[loopCount] != B[loopCount]  
            Result = "False"  
        else  
            Result = "True"  
        loopCount = loopCount + 1  
    }  
    return Result  
}
```

Figure 2. A simple algorithm

- ☒ a. True
- ☐ b. False
- ☐ c. Nil as the execution will be abnormally terminated.



In the execution of this also $A = [4, 3, 7]$, $B = [1, 3, 8]$ and $n = 2$.

Initially Result is set to "True" and loopCount is set to 0

And While loop is executed as follows:

1 iteration, as $A[0] \neq B[0]$, Result is set to "False" and $\text{loopCount} = 0 + 1 = 1$

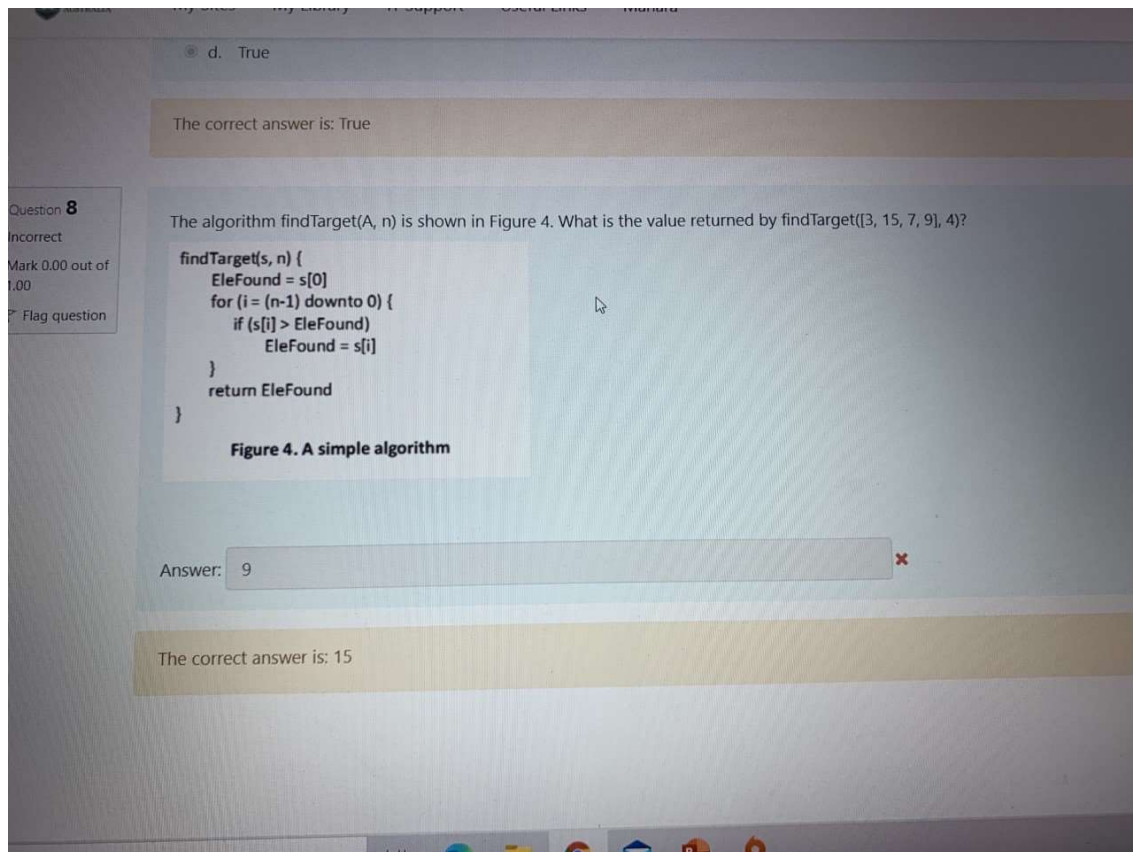
2nd iteration, as $A[1] \neq B[1]$, Result is set to "True" and $\text{loopCount} = 1 + 1 = 2$

Now as the while condition $\text{loopCount} = 2$ Not < 2 so the while loop completed and execute the return statement to return "True"

Q22

Sir, should the right answer is 9 because it is stated from $i = (n-1)$ down to 0?

When the $S(3) = 9 > \text{EleFound}$ which is $S[3] = 0$ then the EleFound will be 9 right? and after that $S[2]$ is not bigger than the EleFound, therefore it should return the last EleFound value which is 9?



Answer:

No.

You need to execute the the algo statement by statement as follows:

Ele Found = $s[0] = 3$

$i = (4-1) = 3$ downto 0

$i = 3$. $s[3] = 9 > \text{EleFound} = 3$, hence, $\text{EleFound} = 9$

$i = 2$. $s[2] = 7$ not $> \text{EleFound} = 9$, hence, EleFound remains at 9

$i = 1$. $s[1] = 15 > \text{EleFound} = 9$, hence, $\text{EleFound} = 15$

$i = 0$, $s[0] = 3$ not > 15 , hence, EleFound remains at 15

return 15

Hence, the answer is 15.

Q23

Why the return value of this Quiz q is 13?

The screenshot shows a Moodle quiz review page. The quiz is titled 'AddMu5(A, n)' and is completed on Tuesday, 13 July 2021, at 7:43 PM. The grade is 2.00 out of 8.00 (25%).

Question 1: The algorithm AddMu5(A, n) is shown below, what is the value returned by AddMu5(3, 1, 2, 4, 5)?

```

AddMu5(A, n) {
    sum = A[0]
    for (i = 1 to (n-1))
        sum = sum + i
    return sum
}

```

Select one:

- ☐ a. 13
- ☐ b. 6
- ☐ c. 15
- ☐ d. 21
- ☒ e. 9

The correct answer is: 13

Question 2: The algorithm AddMu6(A, n) is shown below. What is the value returned by AddMu6(3, 5, 7, 9, 4)?

```

AddMu6(A, n) {
    sum = A[0]
    for (i = 1 to (n-1))
        sum = sum + i
}

```

The page also includes a sidebar with navigation links and a top navigation bar with user information.

>>>>Answer:

First sum is set to A[0]. That is. $\text{sum} = A[0] = 3$.

Then execution of for loop starts from $i = 1$ to $i = 4$ as follows:

1st iteration: $i = 1$ sum is set to current sum (3) +1, that is $\text{sum} = \text{sum} + 1 = 3 + 1 = 4$.

2nd iteration: $i = 2$ sum is set to current sum (4) +2, that is $\text{sum} = \text{sum} + 2 = 4 + 2 = 6$.

3rd iteration: $i = 3$ sum is set to current sum (6) +3, that is $\text{sum} = \text{sum} + 3 = 6 + 3 = 9$.

4th iteration: $i = 4$ sum is set to current sum (9) +4, that is $\text{sum} = \text{sum} + 4 = 9 + 4 = 13$.

Now the for loop execution completes, so it execute the return statement and return $\text{sum} = 13$