***Group No: 3***

***Introduction***

*The program is designed to take in distances from the user and give back the total time for Johnny Redknees to complete the triathlon.*

***Question 1***

*Introduction: functionality of the program*

1. *How to operate the code:*

***The code will only accept values greater than 0, this means the user must enter correct values or the code will not run. The code allows for letters to be input, ensure that only positive numbers are entered. The values that are input will be in meters, make sure these values are entered in meters instead of kilometres.***

*b) Important/useful components of the code*

***The code takes use of lots of functions, I believe this makes the code very easy to edit and understand. The code therefore, has physical areas within the code that deal with certain parts of the output so any further modifications can be made easily.***

*Conclusion:*

*I believe that all of the core functionality of the question have been successfully integrated into the code. Using functions within this application are very useful when repeatedly calculating the same thing (time). This then allowed me to use loops to fill arrays using these function which drastically reduces the clutter within the code.*

***Question 2:***

*This program is designed to take the technical clothing that Johnny can wear to increase his speed and works out which is the fastest combination of clothing for him to wear.*

*Introduction: functionality of the program:*

1. *How to operate the code:*

***The program does function correctly, all of the same input rules apply so the values need to be positive and need to be the values given in metres. The program then outputs all 6 tables (for each combination) and shows the time taken for each part of the triathlon. It then takes the total times for each of those lays them out in a table, however the code does not sort the times, the ordering was done manually. Finally, it outputs the tables to a text document.***

1. *Important/useful components of the code:*

***The code is once again fully modular, each of the calculations for the 3 disciplines are separated clearly. Each part calculates the time taken and then fills arrays to hold the times using loops. Each of the individual functions that are more complex that use functions, loops and arrays have detailed comments above them to indicate what they do as they are harder to follow.***

*Conclusion:*

*To conclude, this question was answered to the highest standard and completed every part of the question apart from the listing in part (c) and successfully identified the best clothing combination (Cycling Shoes and Sunglasses). The code outputs lots of information with fairly minimal code and a good set up and calculations which makes the code feel complete.*

***Question 3:***

*This program is designed to allow for a wider range of input values and scenarios to be input to Johnny’s triathlon. It takes 3 values instead of 1 and applied optimised clothing combination found in question 2.*

*Introduction: functionality of the program:*

1. *How to operate the code:*

***The program uses lots of what we have created in question 1 and 2, therefore it uses the same premise, you enter values of distances and the code output the different times for each of the 3 different distances you input.***

1. *Important/useful components of the code:*

***The code again takes good use of lots of arrays, these can be used to shrink the overall length of the code as these arrays can be filled using loops to then be later recalled using another loop. The code also identifies the shortest distance by comparing each of the 3 lengths, outputting the shortest time using the clothing. Another aspect of the code is within the loop, adding “if (i % 3 == 0){ - if (i > 0)” allows titles etc to be set as separate outputs but within the same loop.***

*Conclusion:*

***In question 3, certain portions of the code produced have been done so in a productive and condensed manner which may need crediting. The use of arrays meant a large portion of code could be shortened using a loop. This became exceptionally useful when producing inputted data, tables and equations. The use of arrays helped exponentially with the final portion of the question, where a short statement about the results needed outputting.***

***Question 4:***

***Overall Conclusion***

*e.g.*

*all/ essential/ core/ a few/ required functions of CW are successfully implemented. There is some initial problems with ….. However later it has been …….*

*use of functions /makes it easy to develop/ cause many problems*

***Word count***