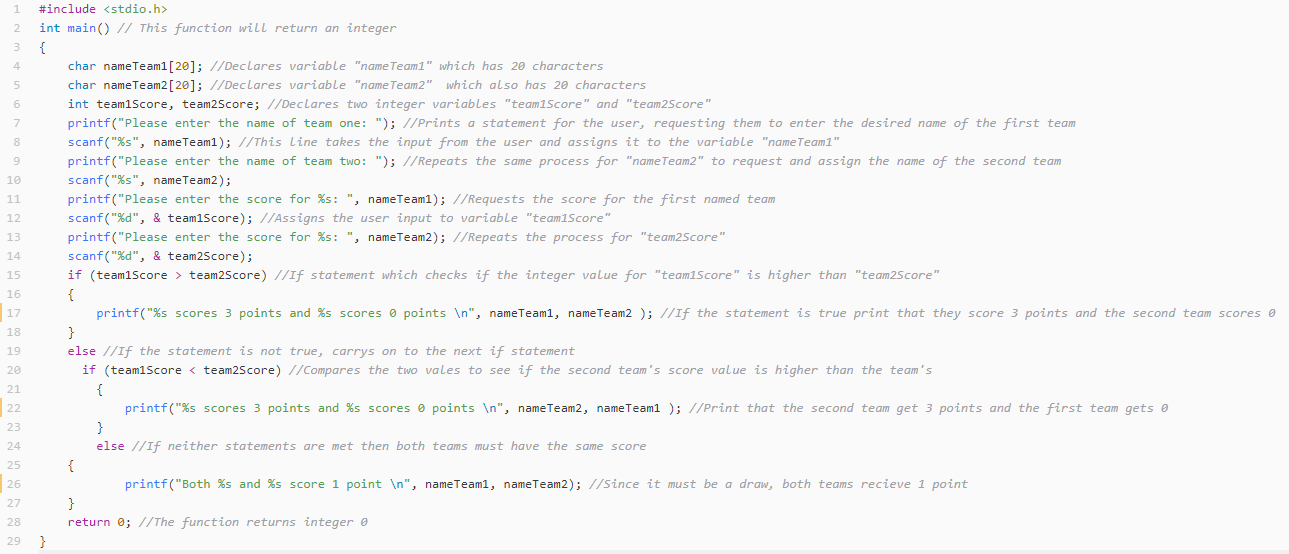
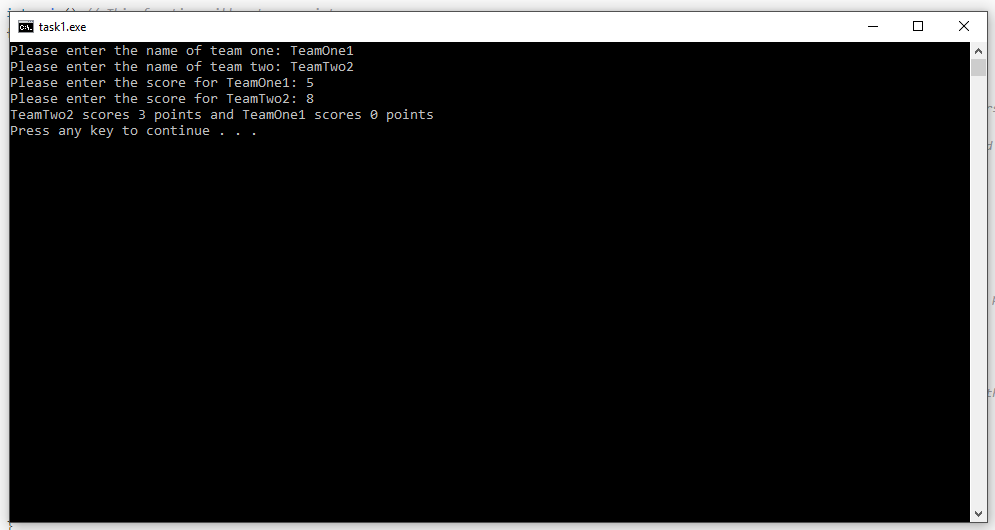
Project Report – Fundamentals of Computing

Tom Penrose

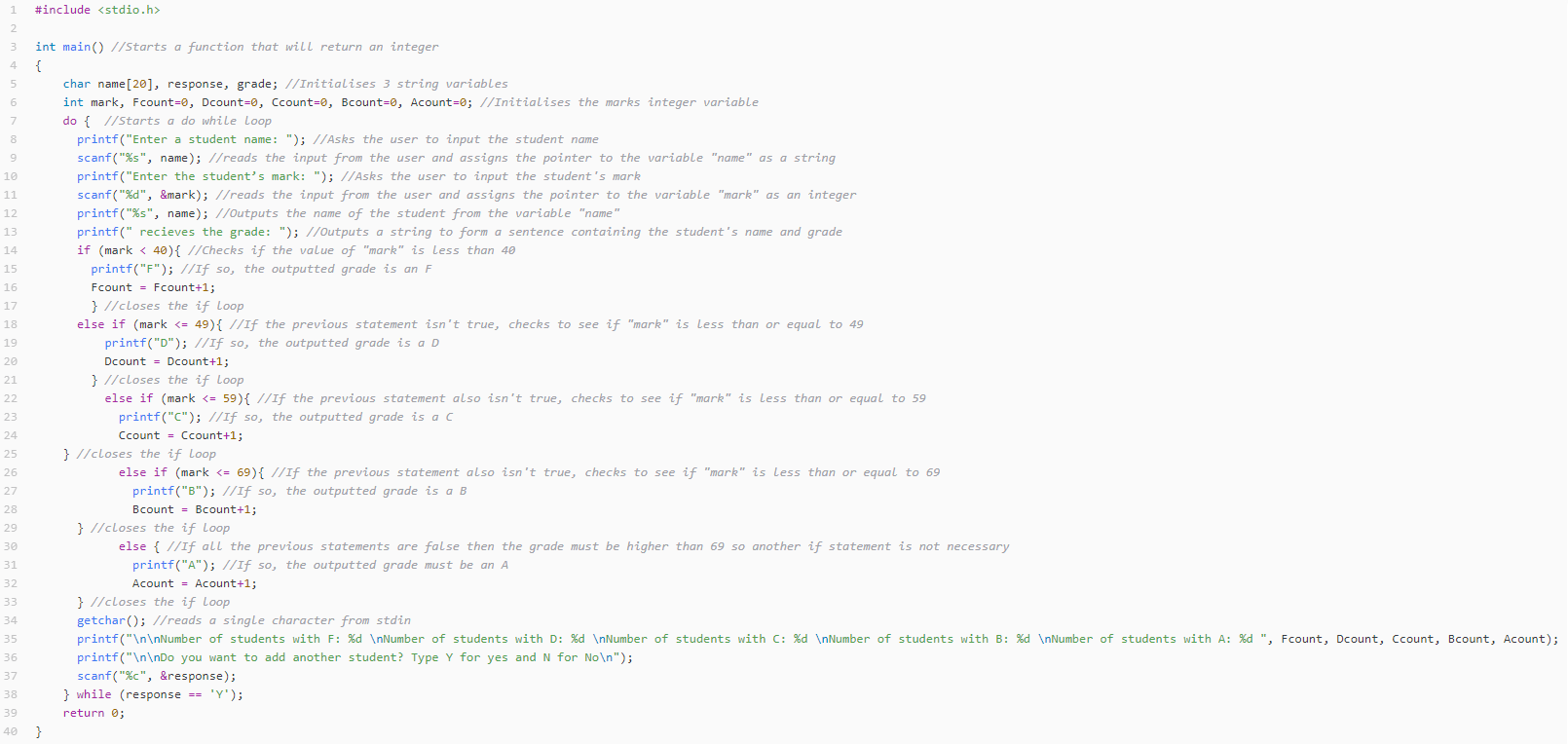
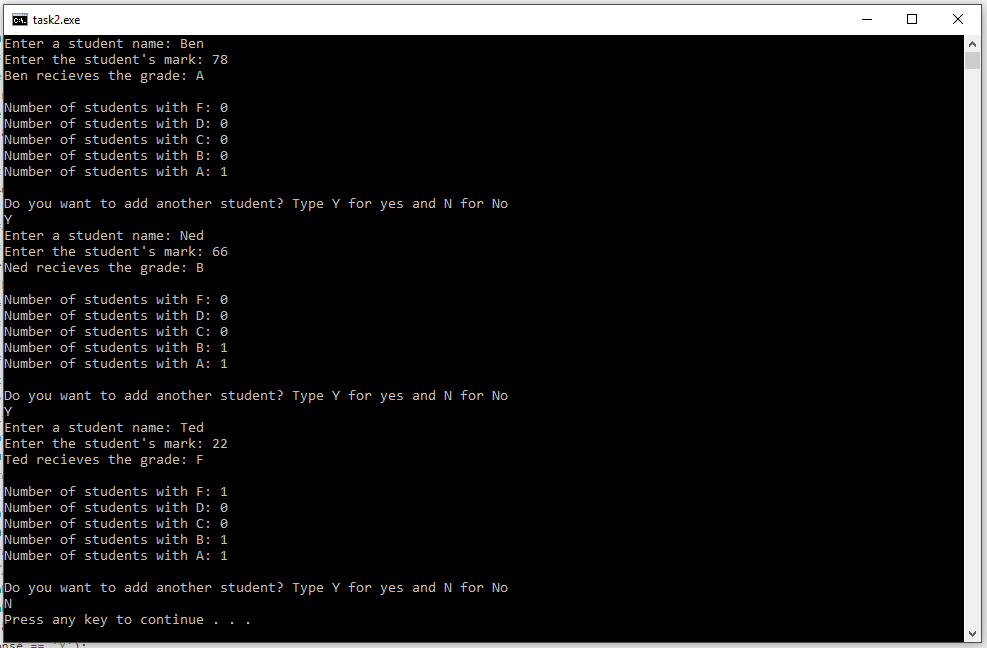
2019 MOD002549 SEM2 F01CAM

2019

# Task 1

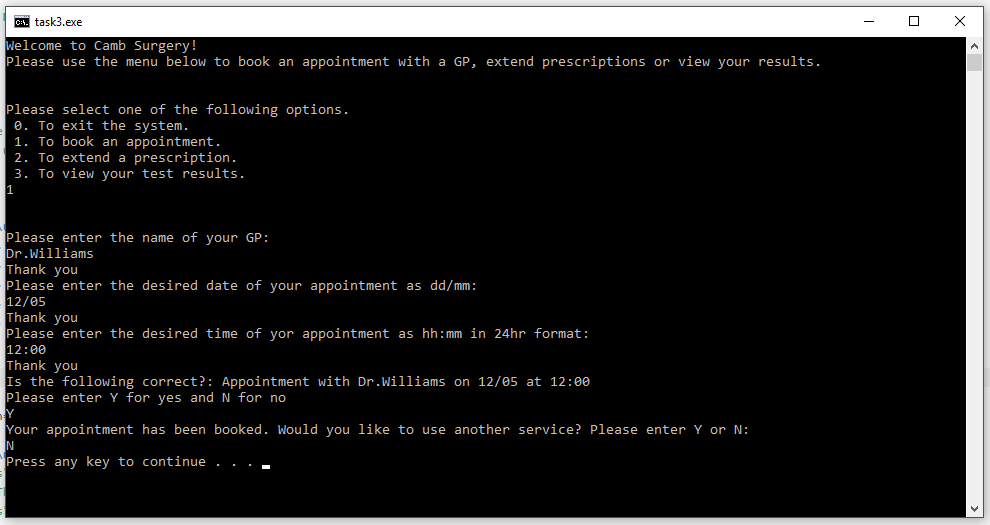
This program takes the names of two teams and the scores for each team. It then compares the scores and prints a different message depending on which team scored more or if it was a draw.

# Task 2

This program takes student’s names and test scores and assigns a grade based on the mark. The code checks the value of “mark” against the grade boundaries to decide which grade to display. It also increments each counter for grades to keep a list of how many of each grade has been achieved by the students. It then asks if more student information will be inputted and loops for the answer ‘Y’.

# Task 3

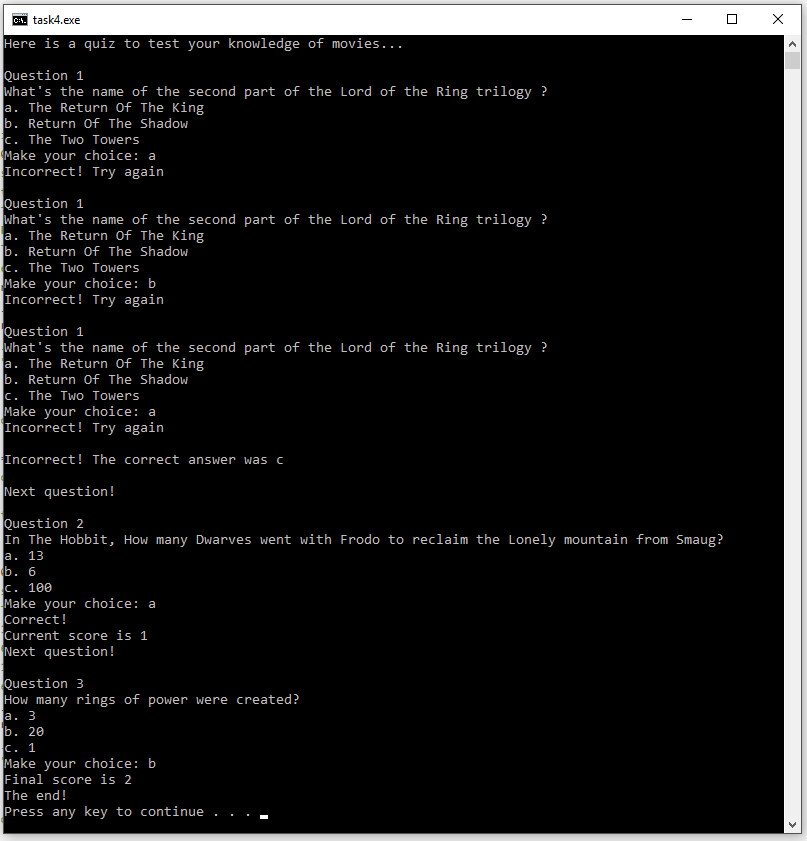
This program displays a menu for 4 options, exit, booking an appointment, extending a prescription and viewing test results. When selected, the 0th option exits the program with a break command. the 1st option requests the GP’s name, date of appointment and time then repeats it to the user for confirmation. They can then choose to re-input this information or accept it and move on. The program then prompts the user if they’d like to use another service where they can choose to pick from the menu again or close the program with another break. The second option asks for a prescription ID and the length of time they’d like to extend it by in days. It again repeats this information to the user and asks for confirmation before asking to continue with another service. The 3rd option requires the user to input a test ID, asks for confirmation, then “prints” the hypothetical test results and asks if the user would like to use another service again.





# Task 4

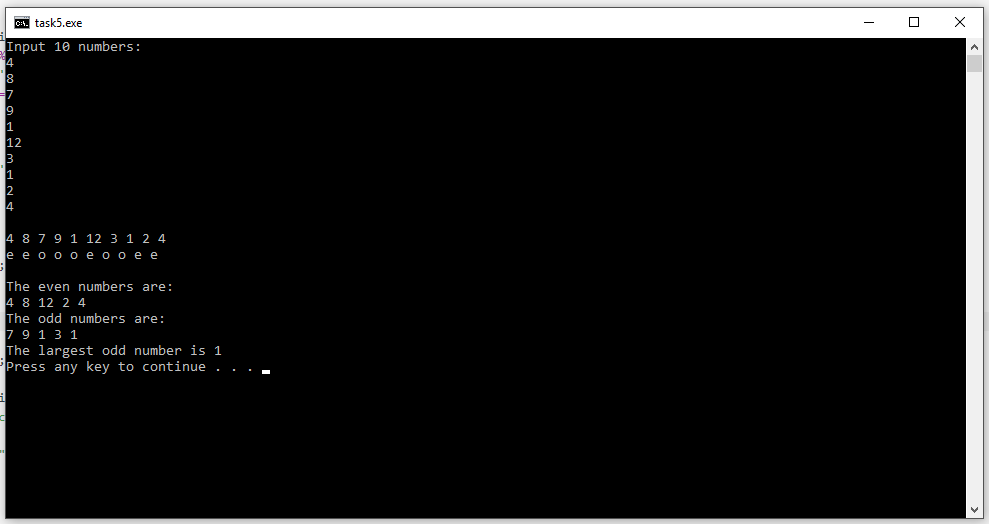
This program is a quiz that gives the user 3 question to answer, picking from 3 possible answers and giving 3 chances to get it correct before moving on. The variable “chances” is set before each question and iterates every time the question is answered. If it reaches 3, the correct answer is revealed and the question is skipped. If the correct answer is entered, the “score” variable is incremented and the next question is displayed. After all 3 questions, the end score is revealed.

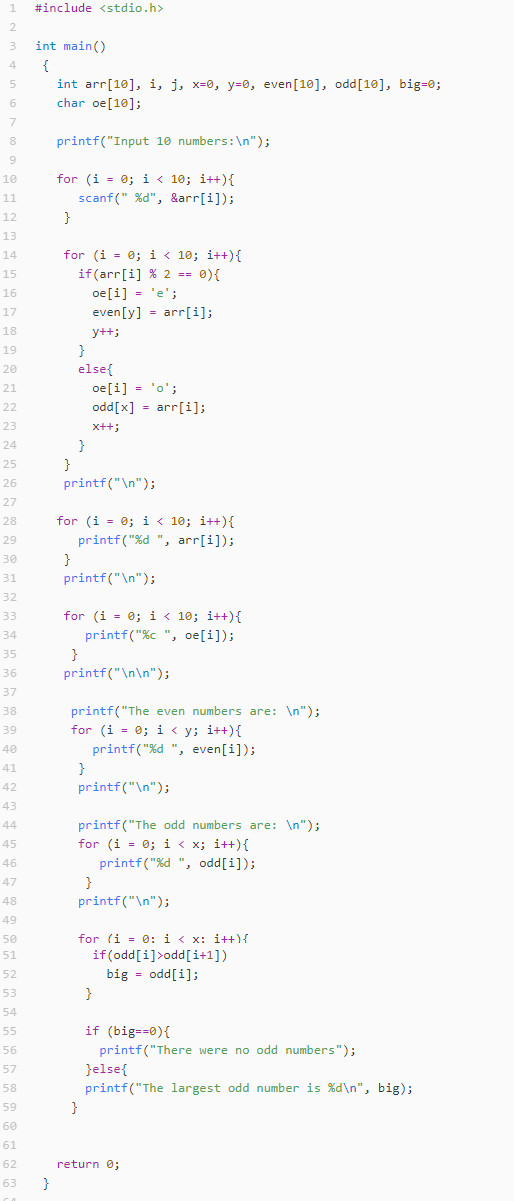




# Task 5

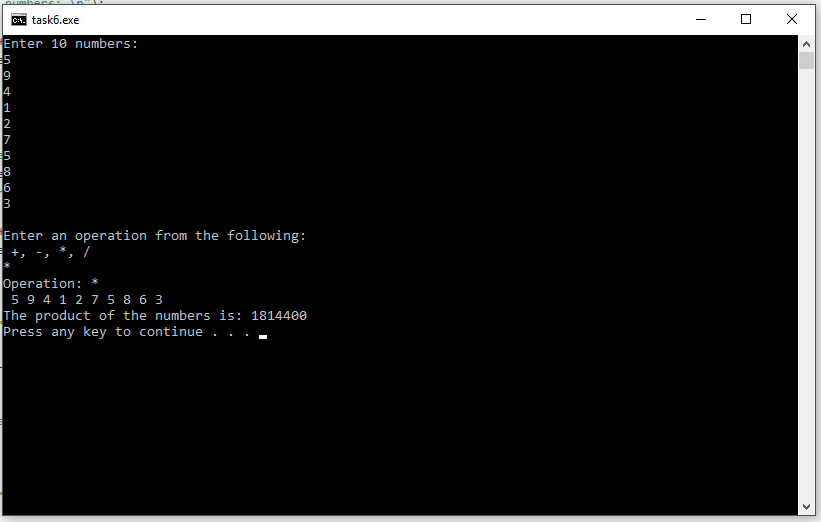
This program finds the largest odd number out of 10 inputted numbers. First, it checks each number in the array to see if it’s odd or even. It prints the numbers and an ‘o’ or ‘e’ to show if they are odd or even. It then sorts them into two separate arrays, one for even numbers and one for odd numbers and displays both arrays. Lastly, it compares every value in the odd array to find the largest value and assigns that to the ‘big’ value which is then printed. If there were no odd numbers, then the value of ‘big’ will be the same as what it was when it was initialised, zero, so it prints that there were no odd numbers.

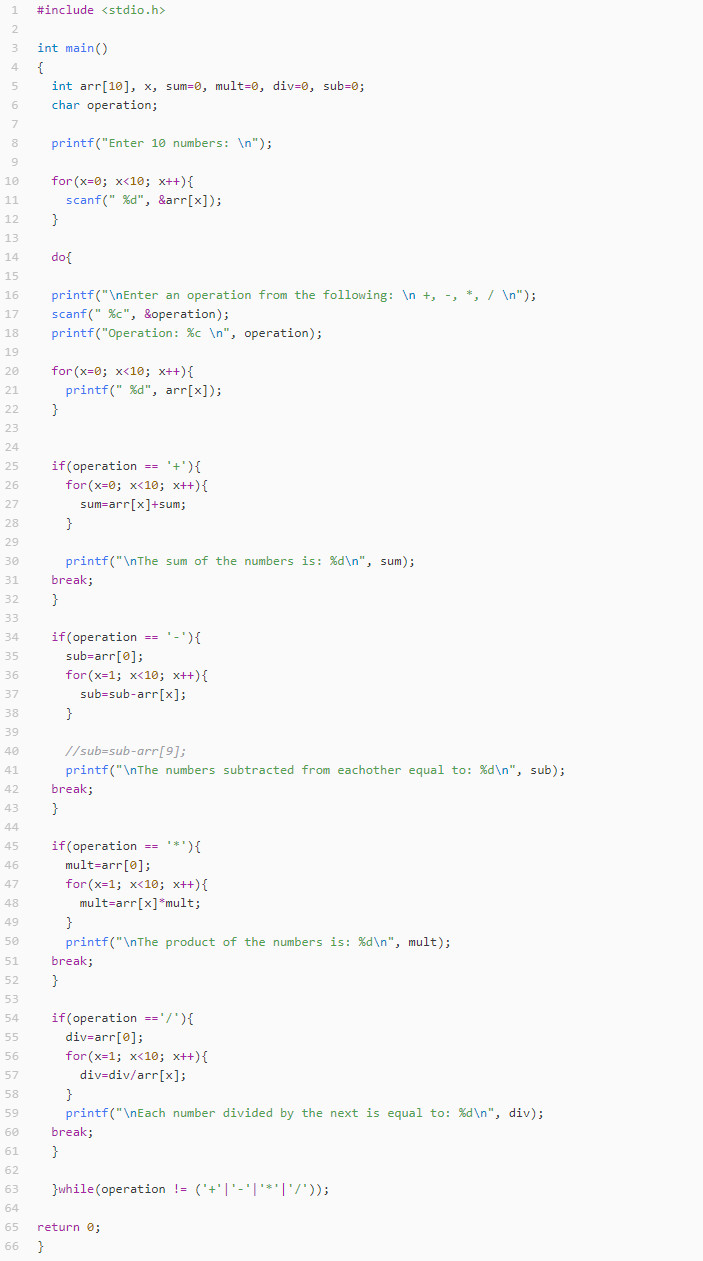




# Task 6

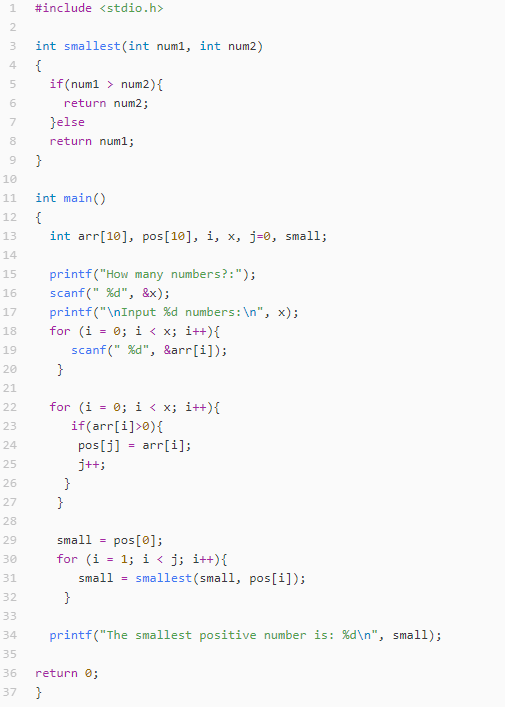
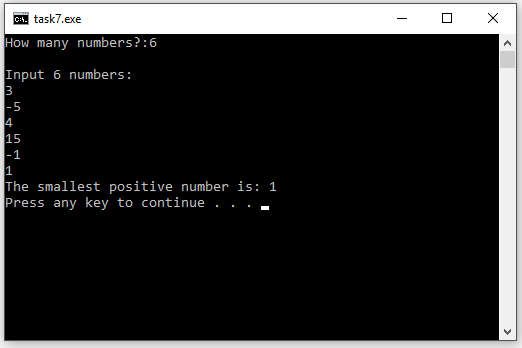
This program takes 10 values then performs an operation on them chosen by the user. First, the 10 values are inputted by the user and stored in an array. The user then enters one of the 4 operators shown in the console and the operation is completed on the values in the array. Depending on the operation, each number from the array is added, subtracted, multiplied or divided from each other to produce the end value which is then printed.





# Task 7

This program prints the smallest possible number from a set of inputted numbers. First, the number of values is inputted so we know how many times to iterate in a later section. Next, the numbers are entered and go into an array. Each value in this array is now checked to see if it’s positive. If so, it goes into a separate array for positive numbers. This array is then fed into the “smallest” function that checks one value against another and returns the smallest. This is repeated for the whole positive array then the smallest positive number is printed.



# Task 8

This program serves as a system to store, process and display student’s marks. First, the main function starts an infinite loop in which it calls the “printMenu” function to display the menu and options. The 1st option runs the function “students” which asks for the user to input the number of students in the class and their names and marks respectively by putting them into two arrays, “name” and “mark”. The 2nd option calls the function “printStudents” which prints out the student’s names and marks from these arrays. The 3rd option calls the “highMark” function which checks each value in the mark array and finds the largest number. It then takes that number and copies the location of the largest value in the mark array and uses it to find the name of that student. It then prints both the name and mark of the student. The 4th option calls the function “highMark” which takes all the student’s marks and creates an average for the class by adding each number together then dividing by the total number of students. It then prints this value. The 5th and final option exits the program with a break.

