

## FixingCodeInProduction

First include the important header file in it then make function named `FixingCodeInProduction` that takes three arguments the sorted array, the size of the array, and the number to find used the binary search algorithm to find the index of an element in a sorted array binary search divides the array into two part, and checks the middle element. If the middle element is equal to the element to find, the index of element is returned. If the middle element is less than the element to find, the algorithm discards the starting half of the array and search in ending half if not then search in starting half it returns the index of the element if found using binary Search and -1 if not found. In the main function after entering the path the code reads the input from user file using `getline` and `ifstream` if there is any error in opening and reading the file it will print the error otherwise it work correctly the first line of the file contains the number to find, and the second line contains the size of the array remaining contains the elements of the array, separated by commas then it prints the array and calls the `FixingCodeInProduction` function to find the index of the number. Finally, the code prints the index of the number.

## GoingOffTheCharts

First include important header file in it then make a `GoingOffTheCharts` function to return the maximum consecutive working hours above 6. Inside the function declare consecutive day and maximum consecutive day (if it above 6) it purpose is to counts the consecutive days using the loop where the element is greater than 6. After the loop it check if consecutive day are greater then maximum and make change according to it. In the main function. In the main function after entering the path

the code reads the input from user file using getline and ifstream if there is any error in opening and reading the file it will print the error otherwise it work correctly its then call the GoingOffTheCharts function to calculate the maximum working hours and print it.

## BribedQueue

Make a function named BribedQueue that calculate the total number of bribes for that we need to sort the array. I used insertion sort because it sort the array in ascending order and perform good for small sized array after each iteration and return the bribes. In the main function after entering the path the code reads the input from user file using getline and ifstream if there is any error in opening and reading the file it will print error otherwise it work correctly then call the BribedQueue function in main to print the total number of bribes taken in the queue.

## FindMedian

Include the important header file in it then make the function name Bubble sort used the bubble sort because it repeat the steps until the array is sorted, compares the elements and swaps them if they are not the correct position until the whole array is sorted in ascending order then I make another function name FindMedian and call bubble sort in it to sort the array and return the middle index. In the main function after entering the path the code reads the input from user file using getline and ifstream if there is any error in opening and reading the file it will print error otherwise it work correctly then call the FindMedian function for finding median value and print it.

