|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Topics: Week 11- 14** | | **Total Marks: 60** | | **Obtained Marks:** | | |
| **Registration No. :** L1F24BSCS0938 | | | **Date:** 28-1-2025 | | | |
| **Section:** A20 | | | **Name:** Talha Khurram | | | |
| **All Questions Attempted (Y/N):** | | | | | | |
| **CLO #** | **CLO Statement** | | | | **Taxonomy Level** | **PLO** |
| **3** | **Design and implement real-world problems using selection statements, loops, and one-dimensional arrays in C++.** | | | | **C3 (Apply)** | **3** |
| **Submission:**  **Soft Copy Deadline: 02-02-2025 till 11:59 PM via Portal**  **Hard Copy Deadline: 03-02-2025**  **Late Submission Policy:**   * 10% Deduction /24 Hours.   **Submission Marks:**   * 5 marks are for submission.   **Instructions: (5 marks will be deducted for not following the instructions)**   * This is an individual assignment. Viva can be conducted OR a Quiz will be taken on the basis of assignment in the next week. * Attempt all questions in sequence. Attach this title page as a front page of assignment. * Assignment should be handwritten/printed on A4 sized page. **(No pages from register please.)** * Submit Hard copy in class and scanned copy of solved assignment on the portal **(Both copies should be submitted before deadline)** | | | | | | |



**UNIVERSITY OF CENTRAL PUNJAB**

**FALL 2020**

FALL 2024

**Course Title: Introduction to Computing**

**Course Code: CP103**

**Assignment No. 4**

**Rubrics for Assignment Evaluation:**

Here are the criteria mentioned below for your assessment evaluation. Give it a read before attempt the assignment. You should read it properly for securing good marks.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Rubrics Detail/ Rubrics Criteria** | **Above Average** | **Sufficient** | **Developing** | **Needs Improvement** | **Marks Distribution** | **Obtained Marks** | |
| **Presentation of Assignment** (The use of front page, page borders, page no’s, table of contents, indentation, use of diagrams/ tables where required). | Excellent presentation, proper use of formatting. | Good presentation, minor issues. | Basic presentation, lacks organization. | Poor formatting, lacks clarity. | Above average = 3 marks  Sufficient = 2 marks  Developing = 1.5 marks  Needs improvement = 1 mark | **Q1** |  |
| **Q2** |  |
| **Q3** |  |
| **Assignment structuring** (proper use of headings & sub-headings, use of bullets & keywords where required and proper division according to the questions asked in assignment). | Excellent structure, clearly divided sections. | Well-structured, minor issues in layout. | Adequately structured but some parts unclear. | Poor structure, lacks organization. | Above average= 2 marks  Sufficient = 1.5 marks  Developing = 1 marks  Needs improvement = 0.5 mark | **Q1** |  |
| **Q2** |  |
| **Q3** |  |
| **Program Logic** (How well you design the solution to program logically. Right use of programming concepts). | Logical and well-thought-out solution. | Good solution, minor issues in logic. | Basic logic, some errors in understanding. | Poor logic, major conceptual issues. | Above average= 5 marks  Sufficient = 3.5 marks  Developing = 2.5 marks  Needs improvement = 2 marks | **Q1** |  |
| **Q2** |  |
| **Q3** |  |
| **Program Ethics** (Use of proper naming conventions in program, indentation & no. of lines) | Excellent naming, formatting, and code structure. | Good adherence to coding practices, minor flaws. | Adequate formatting, naming, or structure. | Poor adherence to coding standards. | Above average= 5 marks  Sufficient = 3.5 marks  Developing = 2.5 marks  Needs improvement = 2 marks | **Q1** |  |
| **Q2** |  |
| **Q3** |  |
| **Correct Output** (Correction of program output, does the program implementing the things for which it was designed?) | Fully correct output, implements all tasks. | Correct output, minor issues in functionality. | Partially correct output, incomplete functionality. | Incorrect output, does not meet requirements. | Above average= 5 marks  Sufficient = 3.5 marks  Developing = 2.5 marks  Needs improvement = 2 marks | **Q1** |  |
| **Q2** |  |
| **Q3** |  |

Introduction to Computing ( Fall 2024 ) - Assignment 4

Question 1 -------------------------------------------------------------------- ( 2 – 3 )

Question 2 -------------------------------------------------------------------- ( 4 – 5 )

Question 3 -------------------------------------------------------------------- ( 6 – 9 )

**Question 1**

**Program :**

#include <iostream>

using namespace std;

int main() {

  const int MAX\_ITEMS = 20;  // maximum number of ids

  int nofitems = 0;

  int actionscount = 0;

  int arrayofids[MAX\_ITEMS] = {0};

  cout << "Enter the number of items in the warehouse (between 5 and 20): ";

  cin >> nofitems;

  if (nofitems >= 5 && nofitems <= 20) {

    cout << "Enter the item IDs: ";

    for (int id = 0; id < nofitems; id++) {

      cin >> arrayofids[id];

    }

    for (int passes = 0; passes < nofitems; passes++) {

      for (int iter = 0; iter < nofitems - passes - 1; iter++) {

        actionscount++;  // counting 1 action for comparison

        if (arrayofids[iter] < arrayofids[iter + 1]) {

          int temp = arrayofids[iter];

          arrayofids[iter] = arrayofids[iter + 1];

          arrayofids[iter + 1] = temp;

          actionscount += 3;  // counting 3 actions for swapping

        }

      }

    }

    cout << "Sorted Item IDs: ";

    for (int sortid = 0; sortid < nofitems; sortid++) {

      cout << arrayofids[sortid];

      if (sortid != nofitems - 1) cout << ", ";  // checking for last comma

    }

    cout << endl;

    cout << "Total actions performed: " << actionscount << endl;

  } else {

    cout << "Invalid input. Please enter a value between 5 and 20. " << endl;

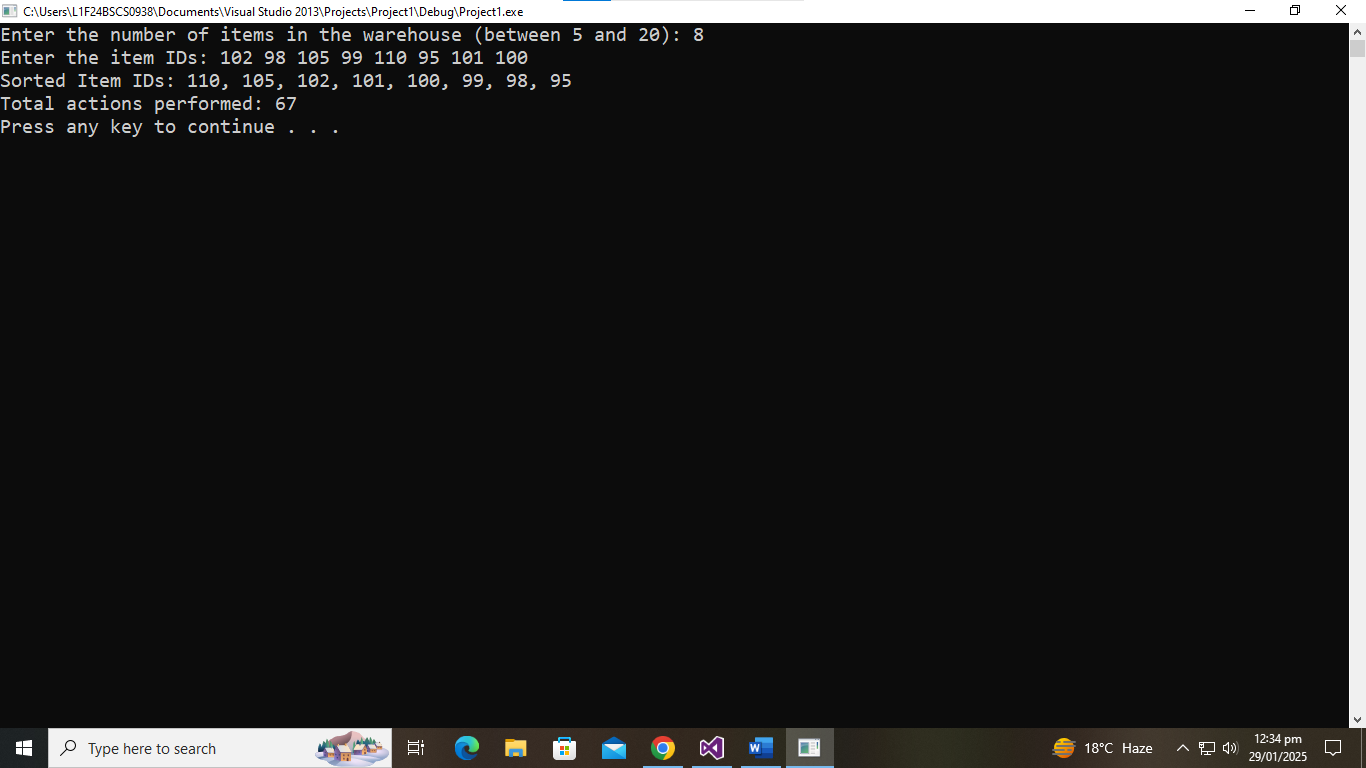
  }

  system("pause");

  return 0;

}

**Output :**

****

**Question 2**

**Program :**

#include <iostream>

using namespace std;

int main() {

    int arrofids[] = {105, 94, 84, 75, 69, 65, 56, 43, 34, 2};

    int itemid;

    int totalids = sizeof(arrofids)/sizeof(arrofids[0]);

    cout << "Enter item ID to search: ";

    cin >> itemid;

    int start=0, end=totalids-1 ,loc=-1, mid;

    int comparisoncount = 0;

    while (start <= end){

        mid = (start + end)/2;

        comparisoncount++; // increasing count on each comparison

        if (arrofids[mid] == itemid) {

            loc = mid;

            break;

        } else if (arrofids[mid] < itemid) {

            end = mid - 1;

        } else {

            start = mid + 1;

        }

    }

    if (loc == -1) {

        cout << "Item not found.  " << endl;

    } else {

        cout << "Item found at index " << loc << endl;

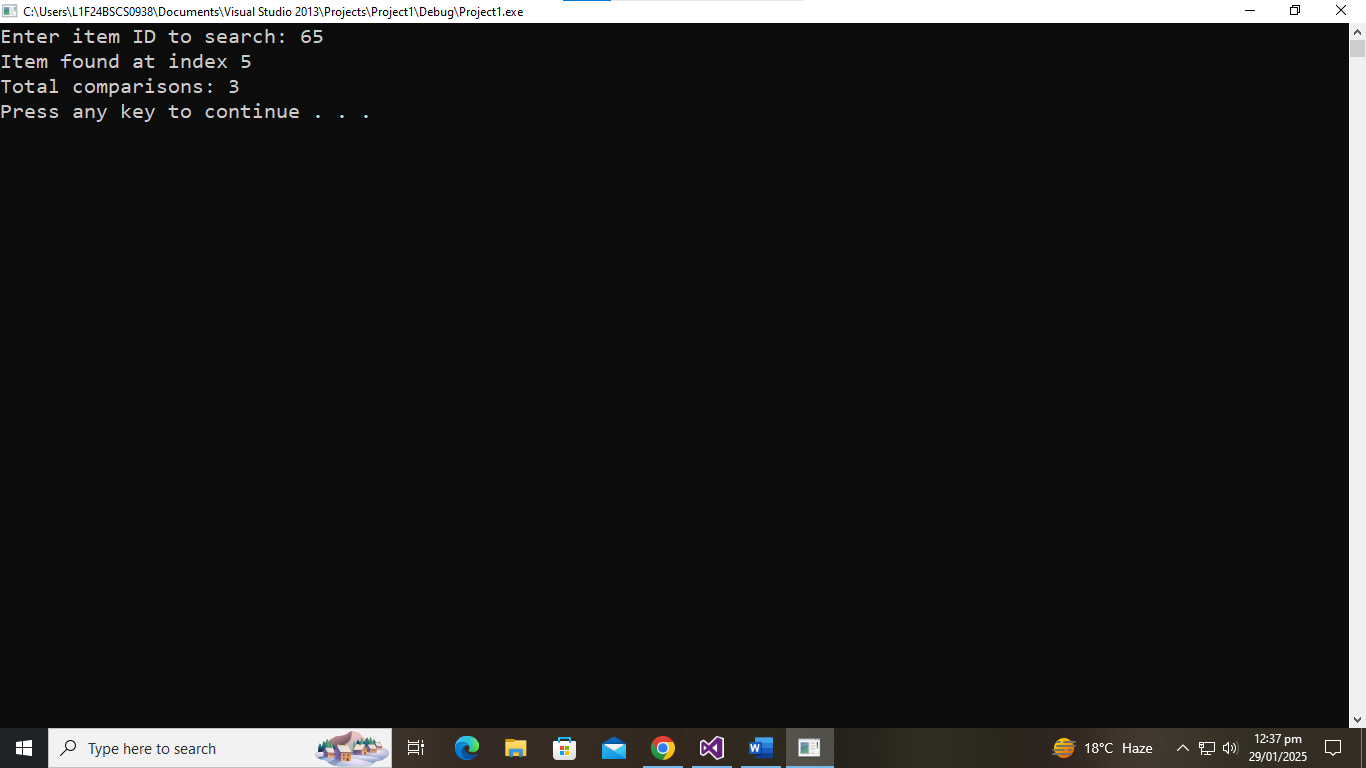
    }

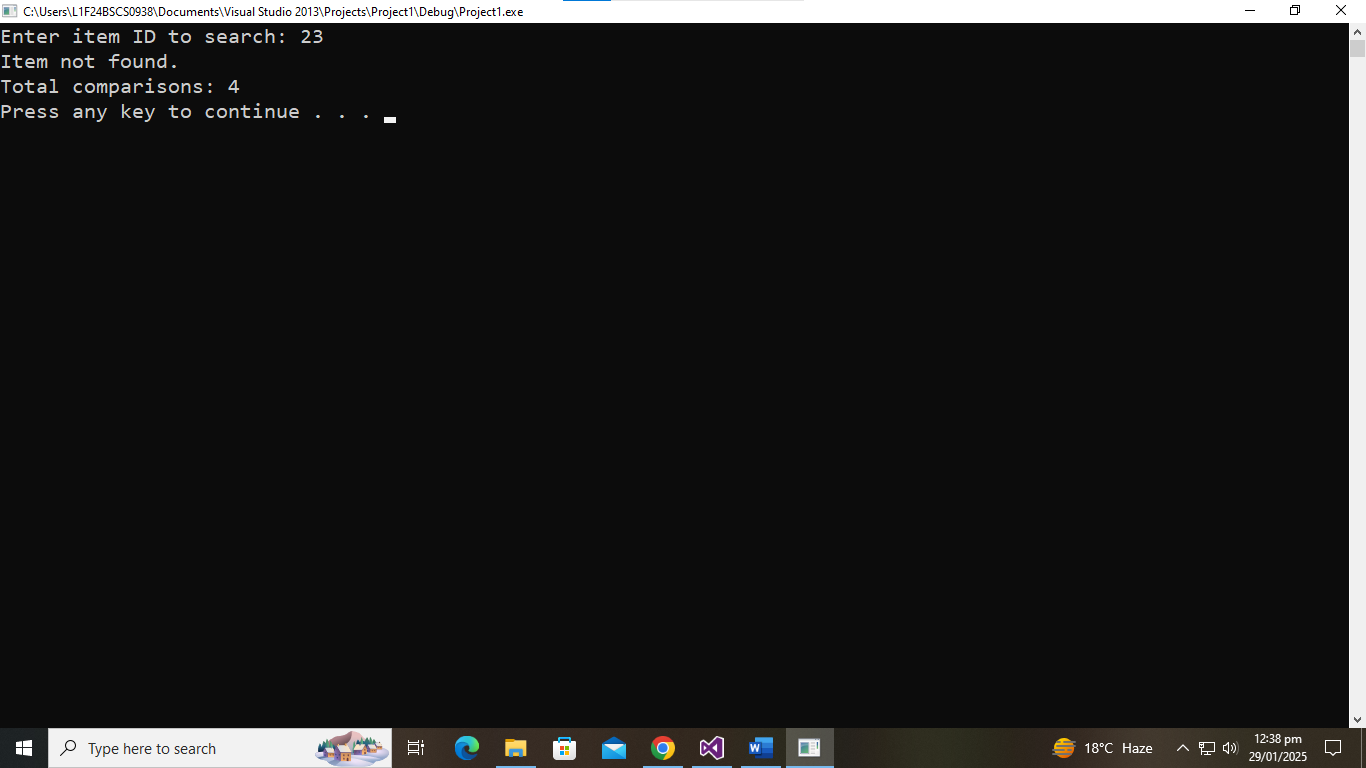
    cout << "Total comparisons: " << comparisoncount << endl;

    return 0;

}

**Output :**

****



**Question 3**

**Program :**

#include <cstring>

#include <iostream>

using namespace std;

int main() {

char approvedusernames[10][50] = {"manager1","manager2","manager3", "supervisor1","supervisor2", "supervisor3", "workerA",  "workerB","workerC","user1"}; // using a 2D array to store usernames bcz there  is no other way to do it

  char enteredusername[50];

  bool isValidusername = false;

  int lengthofappusers = sizeof(approvedusernames) / sizeof(approvedusernames[0]);

  while (!isValidusername) {

    cout << "Enter username: ";

    cin >> enteredusername;

    for (int i = 0; i < lengthofappusers; i++) {

      bool isMatched = true;

      for (int j = 0;

           approvedusernames[i][j] != '\0' || enteredusername[j] != '\0'; j++) {

        if (tolower(approvedusernames[i][j]) != tolower(enteredusername[j])) {

          isMatched = false;

          break;

        }

      }

if (isMatched) {

        cout << "Login successful. Welcome, " << enteredusername << endl;

        isValidusername = true;

        break;

      }

    }

if (!isValidusername) {

      cout << "Invalid username. Please try again" << endl;

    }

    cout << endl;

  }

  bool isValidPassword = false;

  char password[8];

  while (!isValidPassword) {

    cout << "Enter password: ";

    cin >> password;

    int paslength = 0;

    bool isUpper = false, isLower = false, isDigit = false, isspecch = false;

    for (int i = 0; password[i] != '\0'; i++) {

      paslength++;

      if (password[i] >= 'A' && password[i] <= 'Z') isUpper = true;

      if (password[i] >= 'a' && password[i] <= 'z') isLower = true;

      if (password[i] >= '0' && password[i] <= '9') isDigit = true;

      if (password[i] == '@' || password[i] == '#' || password[i] == '\_' ||

          password[i] == '!')

        isspecch = true;

    }

    if (paslength >= 8 && isUpper && isLower && isDigit && isspecch) {

      cout << "Password is strong" << endl;

      isValidPassword = true;

    } else {

cout << "Password is weak. Please include at least one uppercase letter, one lowercase letter, one digit, one special character, and make sure it\’s at least 8 characters long " << endl;

    }

    cout << endl;

  }

char response[4];

  cout << "Would you like to provide feedback? (Yes/No): ";

  cin >> response;

char feedback[100];

  int tchars = 0, twords = 0;

  bool containsgood = false;

  if (strcmp(response, "Yes") == 0) {

    cout << "Enter your feedback: ";

    cin.ignore();                // for clearing the previous input

    cin.getline(feedback, 101);  // using getline to get sentence input from user bcz it cannot be done in any other way

    cout << endl;

    for (int ch = 0; feedback[ch] != '\0'; ch++) {

      tchars++;

      if (feedback[ch] == ' ' || feedback[ch] == '\0') twords++;

      if (feedback[ch] == 'g' && feedback[ch + 1] == 'o' &&

          feedback[ch + 2] == 'o' && feedback[ch + 3] == 'd') {

        containsgood = true;

      }

    }

    cout << "Feedback Analysis:  " << endl;

    cout << "Total characters: " << tchars << endl;

    cout << "Total words: " << twords << endl;

    if (containsgood) {

      cout << "Feedback contains the word \"good\"." << endl;

    } else {

      cout << "Feedback not contains the word \"good\"." << endl;

    }

    cout << endl;

  } else {

    cout << "Thank you for using the system!" << endl;

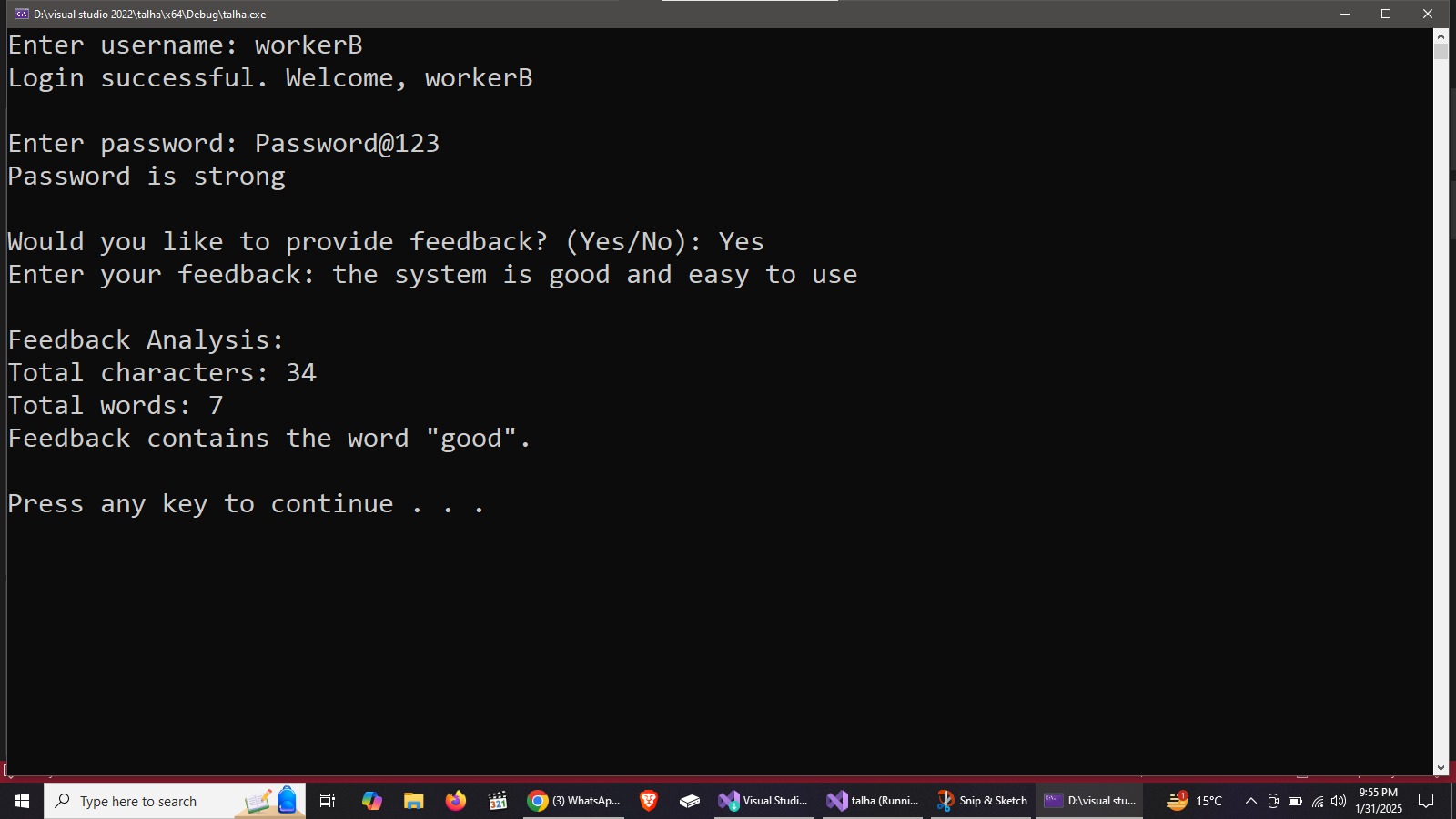
  }

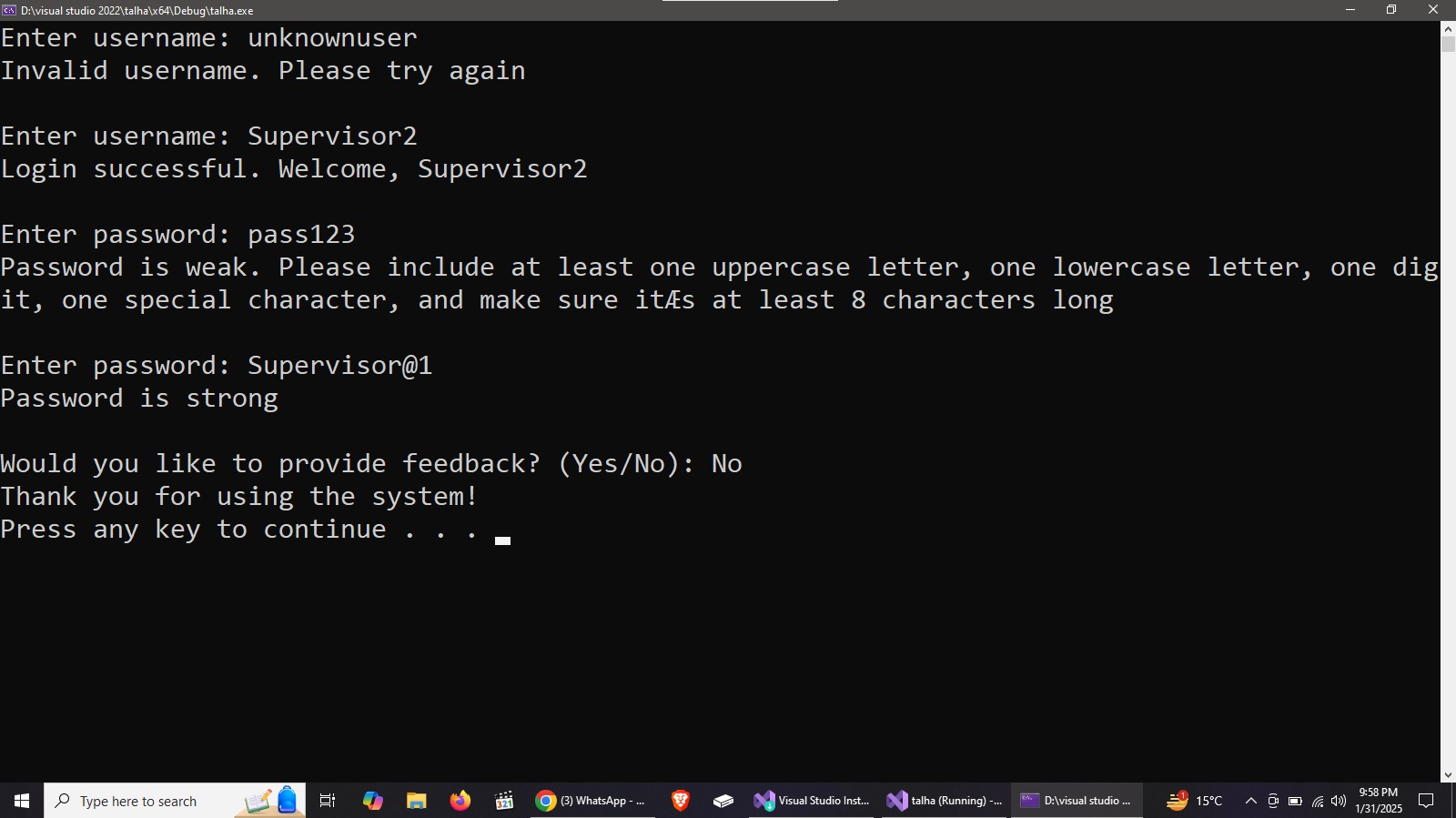
  system("pause");

  return 0;

}

**Output :**

****

****