AIM:

The "Student Grade Checker" is a command-line program designed to assist educators in evaluating and recording student performance based on their academic achievements. This project aims to streamline the grading process, providing a user-friendly interface for entering and calculating grades, as well as generating detailed reports for further analysis.

SCOPE:

1. **Student Information:** The program prompts the user to input the student's name, class, section, and the number of subjects they are enrolled in. This information is crucial for generating accurate reports.
2. **Subject-wise Grading:** For each subject, the program captures the following details:

a) Marks Obtained: The actual score attained by the student.
b) Total Marks: The maximum attainable score for the subject.
c) Credits: The weightage or importance of the subject in the overall assessment.
d) Absent/Present: The program records if the student was absent for the exam in a particular subject.

3. **Absence Handling:** If a student is marked as absent for a subject, the program automatically assigns a credit of 0 for that subject, ensuring that it is excluded from CGPA calculations.
4. **CGPA Calculation:** The Cumulative Grade Point Average (CGPA) is computed based on the subjects for which the student was present. The CGPA is calculated as the total Grade Points divided by the total Credits.
5. **Dynamic Subject Entry:** The program accommodates a flexible number of subjects (up to a maximum of 7) to cater to different curriculum requirements.
6. **Output Report:** Upon entering all necessary details, the program generates a detailed report in a text file format. The report includes the student's name, class, section, the list of subjects with associated details (Marks Obtained, Total Marks, Credits, Absent/Present), and the computed CGPA.

MODULES:

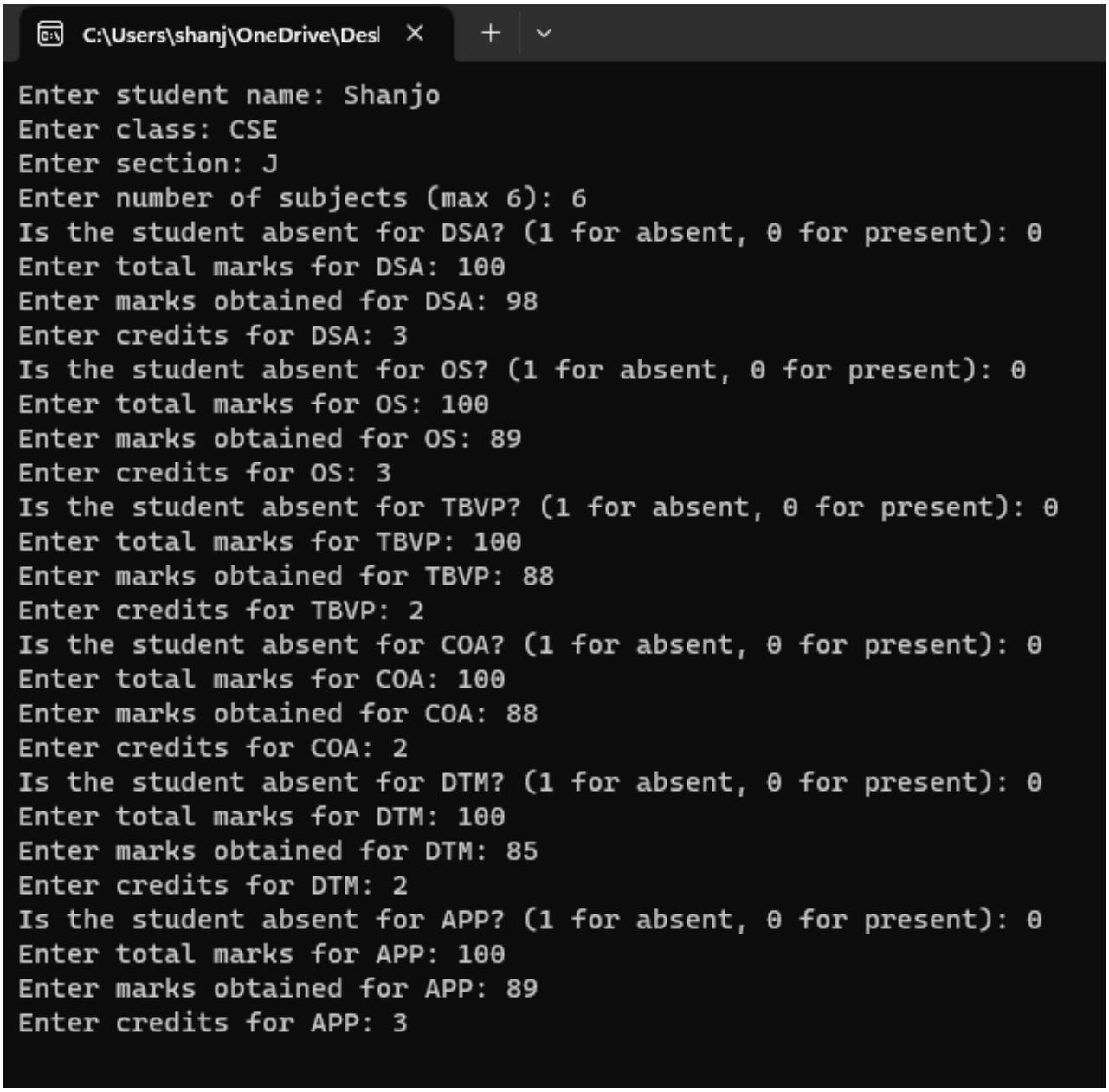
1. **Purposeful Grading Tool:** The project serves as a purposeful tool for educators to efficiently evaluate and record student performance based on their academic achievements.
2. **User-Friendly Interface:** It provides a user-friendly command-line interface, making it accessible and easy to navigate for teachers and administrators.
3. **Comprehensive Student Information:** The program captures vital student details, including name, class, section, and the number of subjects they are enrolled in, ensuring accurate record-keeping.
4. **Flexible Subject Entry:** It allows for a dynamic number of subjects to cater to various curriculum requirements, ensuring adaptability to different educational settings.
5. **Subject Details Handling:** For each subject, the program records specific details such as marks obtained, total marks, credits, and attendance status, providing a comprehensive overview of the student's performance.
6. **Attendance Consideration:** It intelligently handles situations where a student is marked as absent for an exam, ensuring that such instances do not impact the overall grade calculation.
7. **CGPA Computation:** The program automates the calculation of the Cumulative Grade Point Average (CGPA) based on the subjects the student has attended, providing an accurate measure of overall performance.
8. **Report Generation:** Upon inputting all necessary data, the program generates a detailed report in text file format, containing essential information about the student's grades, attendance, and CGPA.
9. **Adaptability to Different Educational Institutions:** The project's flexible design allows it to be used by a wide range of educational institutions, from schools to universities, catering to diverse grading systems and structures.
10. **Potential for Future Enhancements:** While currently fully functional, the project has room for potential enhancements, including the addition of a graphical user interface, database integration, and customizable grading scales, to further meet the evolving needs of educators.

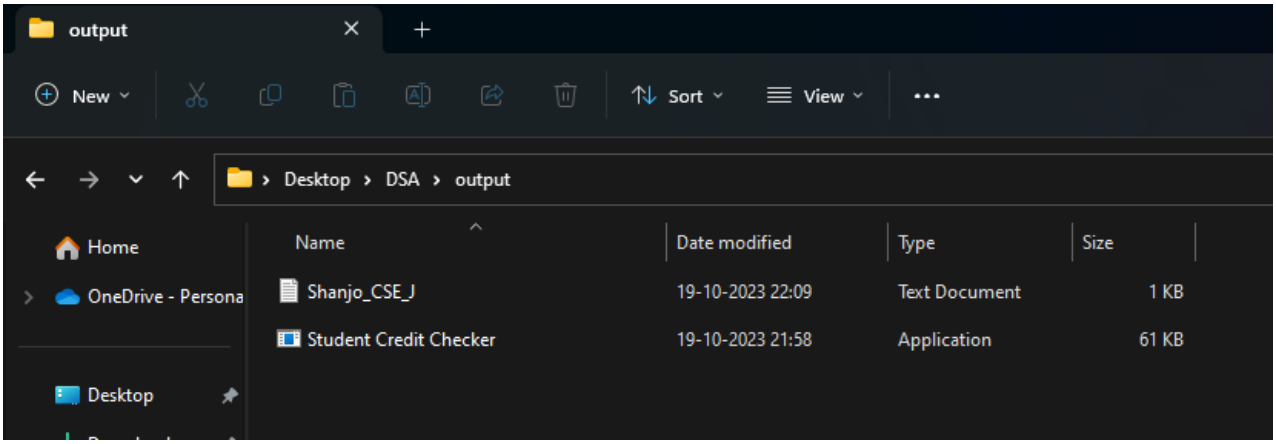
CODE:

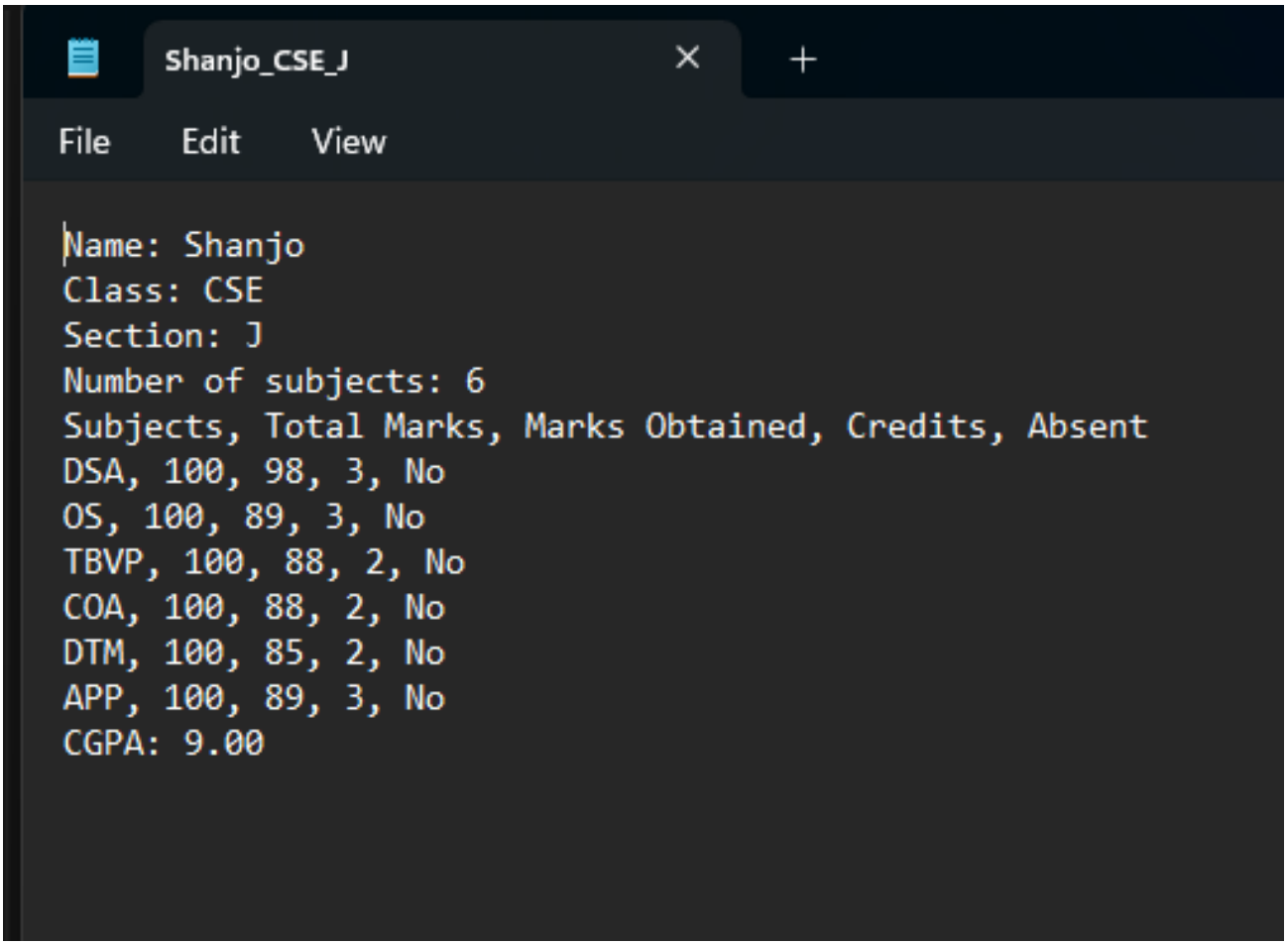
```
#include <stdio.h>
#define MAX_SUBJECTS 6
const char* SUBJECT_NAMES[MAX_SUBJECTS] = {"DSA", "OS", "TBVP",
"COA", "DTM", "APP"};
int main() {
   char name[100], class[10], section[10];
   int numSubjects;
   printf("Enter student name: ");
   scanf("%s", name);
   printf("Enter class: ");
   scanf("%s", class);
   printf("Enter section: ");
   scanf("%s", section);
   printf("Enter number of subjects (max %d): ", MAX_SUBJECTS);
   scanf("%d", &numSubjects);
   if (numSubjects <= 0 || numSubjects > MAX_SUBJECTS) {
      printf("Invalid number of subjects. Please enter a number between 1 and %d.\n",
MAX_SUBJECTS);
      return 1;
   }
   int marks[MAX_SUBJECTS], credits[MAX_SUBJECTS],
absent[MAX_SUBJECTS], totalMarks[MAX_SUBJECTS];
   for (int i = 0; i < numSubjects; i++) {
      printf("Is the student absent for %s? (1 for absent, 0 for present): ",
SUBJECT_NAMES[i]);
      scanf("%d", &absent[i]);
      if (absent[i] != 1 && absent[i] != 0) {
         printf("Invalid input for absence. Please re-enter.\n");
         i--;
         continue;
      }

      if (!absent[i]) {
         printf("Enter total marks for %s: ", SUBJECT_NAMES[i]);
         scanf("%d", &totalMarks[i]);
         if (totalMarks[i] <= 0 || totalMarks[i] > 100) {
            printf("Invalid total marks. Please enter a number between 1 and 100.\n");
            i--;
            continue;
         }
         printf("Enter marks obtained for %s: ", SUBJECT_NAMES[i]);
         scanf("%d", &marks[i]);
```

```
            if (marks[i] < 0 || marks[i] > totalMarks[i]) {
                printf("Invalid marks. Please enter a number between 0 and %d.\n",
totalMarks[i]);
                continue;
                i--;
            }
            printf("Enter credits for %s: ", SUBJECT_NAMES[i]);
            scanf("%d", &credits[i]);

            if (credits[i] < 0) {
                printf("Invalid credits. Please enter a positive number.\n");
                i--;
                continue;
            }
        } else {
            credits[i] = 0;
        }
    }
    float totalCredits = 0, totalGradePoints = 0;

    for (int i = 0; i < numSubjects; i++) {
        totalCredits += credits[i];
        totalGradePoints += ((float)marks[i] / totalMarks[i]) * credits[i];
    }

    float cgpa = (totalCredits != 0) ? totalGradePoints / totalCredits*10 : 0;

    char file_name[150];
    sprintf(file_name, "%s_%s_%s.txt", name, class, section);

    FILE *file = fopen(file_name, "w");
    if (file != NULL) {
        fprintf(file, "Name: %s\nClass: %s\nSection: %s\n", name, class, section);
        fprintf(file, "Number of subjects: %d\n", numSubjects);
        fprintf(file, "Subjects, Total Marks, Marks Obtained, Credits, Absent\n");
        for (int i = 0; i < numSubjects; i++) {
            fprintf(file, "%s, %d, %d, %d, %s\n", SUBJECT_NAMES[i], totalMarks[i],
marks[i], credits[i], (absent[i] == 1) ? "Yes" : "No");
        }
        fprintf(file, "CGPA: %.2f\n", cgpa);
        fclose(file);
        printf("Student information saved to %s\n", file_name);
    } else {
        printf("Error saving file.\n");
    }
    return 0;}
```

OUTPUT:


C:\Users\shanj\OneDrive\Des
Enter student name: Shanjo
Enter class: CSE
Enter section: J
Enter number of subjects (max 6): 6
Is the student absent for DSA? (1 for absent, 0 for present): 0
Enter total marks for DSA: 100
Enter marks obtained for DSA: 98
Enter credits for DSA: 3
Is the student absent for OS? (1 for absent, 0 for present): 0
Enter total marks for OS: 100
Enter marks obtained for OS: 89
Enter credits for OS: 3
Is the student absent for TBVP? (1 for absent, 0 for present): 0
Enter total marks for TBVP: 100
Enter marks obtained for TBVP: 88
Enter credits for TBVP: 2
Is the student absent for COA? (1 for absent, 0 for present): 0
Enter total marks for COA: 100
Enter marks obtained for COA: 88
Enter credits for COA: 2
Is the student absent for DTM? (1 for absent, 0 for present): 0
Enter total marks for DTM: 100
Enter marks obtained for DTM: 85
Enter credits for DTM: 2
Is the student absent for APP? (1 for absent, 0 for present): 0
Enter total marks for APP: 100
Enter marks obtained for APP: 89
Enter credits for APP: 3


output
New
Sort
View
Desktop › DSA › output
Home
OneDrive - Persona
Desktop
Name
Date modified
Type
Size
Shanjo_CSE_J
19-10-2023 22:09
Text Document
1 KB
Student Credit Checker
19-10-2023 21:58
Application
61 KB



CONCLUSION:

The Program to calculate the Grade for Students is Successfully executed and the EXE file is Created, We can do further developments after by reconstructing this code.