

DOCUMENTATION

- 1) The assignment consists of 2 tasks:
 - i) Read Student ID and print out his/her name, ID, number of questions missed, Lists of miss questions, percentage, and grade.
 - ii) Generate a report on a output file which contain all students' name, ID, percentage and grade.

Task 1

- 1) Functions involved:

```
18  /*****PROGRAM FUNCTIONS*****/
19  void readFile(char ID[]);
20  void identity(char name[], char matric[]);
21  void compareAnswers(char data[], int flag, FILE* file);
22  void printMissQuestion(float no[], char wrong[], char correct[]);
23  void Percentage_and_Grade(int error);
```

- 2) Flow:

- i) Main function:

```
177 int main()
178 {
179     char ID[20];
180     int yesno;
181
182     printf("ENTER THE STUDENT ID: ");
183     scanf("%s",&ID);
184     printf("\n\nEXAM RESULT\n\n");
185     readFile(ID);
186 }
```

- The program will get the ID from user.
- The ID will be the input of readFile() function.

ii) readFile():

```

29 char name[14], matrikno[11], Answers[42];
30 char temp[11];
31 int wrong = 0, i;
32 FILE *StudentAnswers;
33
34 temp[0] = ' ';
35 for(i = 1; i <= 11; i++) { //ID syncing for comparison p
36     temp[i] = ID[i-1];
37 }
38
39 //StudentAnswers = fopen("Student_List.txt", "r");
40 StudentAnswers = fopen("StudentAnswers.dat", "r");
41 while(!feof(StudentAnswers)){
42     fgets(name, 14, StudentAnswers);
43     fgets(matrikno, 11, StudentAnswers);
44     fgets(Answers, 42, StudentAnswers);
45     for(i=0; i<11; i++){
46         if(temp[i] != matrikno[i]) wrong++;
47     }
48
49     if(wrong == 0){
50         identity(name, matrikno);
51         compareAnswers(Answers, 1, StudentAnswers);
52         break;
53     }
54     wrong = 0; //reset the wrong factor for next stu
55 }
56 fclose(StudentAnswers);
57 }

```

- The function will open the StudentAnswers.dat file, read the data in the file until the end of file is reached. First, the function will get Student name store in “name” variable, Student ID store in “matrikno” variable, and his/her Answers store in “Answers” variable.
- To find the student’s name and the Answers corresponding to the input ID, the comparison between the input ID and “matrikno” variable is needed (code line 45 – 54).
- 2 outcomes:
 - a) “matrikno” variable not equal to input ID:

“wrong” variable will +1 if the comparing character between the two variables is not the same.

If wrong variable not equal to 0, the program will proceed to read next student’s name, matrikno and Answers.

- b) "matrikno" variable is equal to input ID:
"wrong" variable = 0, thus the if statement (line 49 – 53) will be executed. The "name" and "matrikno" will be the input of identity() function, and Answers will be the input of compareAnswers() function.

iii) identity():

```
59 void identity(char name[], char matric[]){  
60     printf("Name      : %s\n",name);  
61     printf("Student ID:%s\n",matric);  
62 }
```

- This function will print The name and student ID on screen.

iv) compareAnswers():

```
64 void compareAnswers(char data[], int flag, FILE* file){  
65     char temp[42];  
66     char wrongAns[20], correctAns[20];  
67     int i, k = 0, missed = 0, error;  
68     float number[20];  
69  
70     for(i = 1; i < 42; i++){ //syncing data and Answer Scheme an  
71         temp[i - 1] = data[i];  
72     }  
73  
74     char AnswerScheme[41];  
75     FILE *RealAnswers;  
76     RealAnswers = fopen("Answers.txt", "r");  
77     fgets(AnswerScheme, 42, RealAnswers);  
78  
79     for(i = 0; i < 42; i++){ //compring each charcter of array  
80         if(temp[i] != AnswerScheme[i]) { //wrong ans detecte  
81             number[k] = 0.5 * i + 1;  
82             wrongAns[k] = temp[i];  
83             correctAns[k] = AnswerScheme[i];  
84             missed++;  
85             k++;  
86         }  
87     }  
88  
89     error = missed - 1;  
90     if(flag){  
91         printf("Number of questions missed. : %d\n", error);  
92         printMissQuestion(number, wrongAns, correctAns);  
93         Percentage_and_Grade(error);  
94     }
```

- This function will get the Correct Answers from Answers.txt file (The correct answers stored in "AnswerScheme" variable.

- Then, the function will compare the input “data” array with “AnswerScheme” array element by element (question by question) in a “for” loop.
- 2 outcomes:
 - a) nth question is correct – The loop will proceed to compare with the next question.
 - b) nth question is incorrect – The ‘if’ statement (line 80 – 86) will be executed. The following things will be stored in according variable:
 - Incorrect question list -> “number” array
 - Incorrect answer list -> “wrongAns” array
 - Correct answer of the incorrect answer list -> “correctAns” array

(Example: if the student gets question 2, 6, 11, 18, 20
incorrect: number = {2,6,11,18,20})

- The total number of missed questions will be stored in “error” variable and print on screen.
- Then, the three array mentioned in “b” will be passed to printMissQuestion() function, and “error” variable will be passed to Percentage_and_Grade() function.

v) printMissQuestion():

```
100 void printMissQuestion(float no[], char wrong[], char correct[]){
101     int i = 0;
102
103     printf("List of the questions missed: \n\n");
104     printf("Question      CorrectAnswer      StudentAnswer\n");
105
106     while(1){
107         if(no[i]>20){
108             break;
109         }
110         printf("%.0f\t\t%c\t\t%c\n", no[i], correct[i], wrong[i]);
111         i++;
112     }
113 }
```

- This function will print out the list of the question missed.

vi) Percentage_and_Grade():

```
115 void Percentage_and_Grade(int error){
116     float percentage;
117     int right = 20 - error;
118     char grade;
119
120     percentage = ((float)right/20)*100;
121
122     if(percentag>=80 &&percentage<=100) grade = 'A';
123     else if(percentag>=70 &&percentage<80)grade = 'B';
124     else if(percentag>=60 &&percentage<70)grade = 'C';
125     else grade = 'F';
126
127     printf("\nPercentage: %.2f %, GRED: %c\n",percentage,grade);
128 }
```

- This function will calculate the percentage, assign the grade based on percentage, and print both value on screen.

Task 2

a) Functions involved:

```
void compareAnswers(char data[], int flag, FILE* file);
void PrintGrade(int error, FILE* file);
void printreport();
```

b) Flow:

i) printreport():

```
146 void printreport(){
147
148     char name[14], matrikno[11], Answers[42];
149     char temp[42];
150     int wrong = 0, i;
151     FILE *StudentAnswers, *output;
152
153     //StudentAnswers = fopen("Student_List.txt", "r");
154     StudentAnswers = fopen("StudentAnswers.dat", "r");
155     //output = fopen("output_file.txt", "w");
156     output = fopen("OUTPUT.out", "w");
157     printf("\nStart generating output file...\n"); //testi
158
159     fprintf(output, "LIST OF STUDENTS AND GRADES\n\n");
160     fprintf(output, "NAME\t\t ID\t\tPERCENTAGE\tGRADE\n");
161     while(!feof(StudentAnswers)){
162         fgets(name, 14, StudentAnswers);
163         fprintf(output, "%s\t", name);
164         fgets(matrikno, 11, StudentAnswers);
165         fprintf(output, "%s\t", matrikno);
166         fgets(Answers, 42, StudentAnswers);
167         compareAnswers(Answers, 0, output);
168     }
169     printf("output file successfully generated.\n"); //tes
170     fclose(StudentAnswers);
171     fclose(output);
172 }
```

- This function will open StudentAnswers.dat file and OUTPUT.out file.
- Name, ID and Answers will get from the input file. Answers will be passed to compareAnswers() function (mentioned before), to get the percentage and grade.
- Name, ID, percentage and grade of all students will be written in OUTPUT.out file.