Họ và tên: Đinh Việt Hoàng

Mã sinh viên: 23010051

Bài tập lớn học phần nhập môn lập trình C: Xây dựng một chương trình quản lý sinh viên liên kết với MySQL

Chương trình bao gồm các file:

* main.c
* function.h
* std\_string.h
* type.h

main.c

**#include <stdio.h>**

**#include "function.h"**

**int main() {**

**printf("WELCOME TO THE STUDENT MANAGEMENT PROGRAM\n\n");**

**printf("This program is designed to manage student information.\n\n");**

**printf("Maximize the console for the best experience.\n\n");**

**int choice;**

**do {**

**printf("Enter: 1. To initialize a list to store students\n       2. To exit the program\n       ---->");**

**my\_scanf("%d", &choice);**

**} while (choice != 1 && choice != 2);**

**if (choice == 2) {**

**printf("Exit program successfully.");**

**return 0;**

**}**

**student\_list\* list = (student\_list\*)malloc(sizeof(student\_list));**

**initialize\_student\_list(list, 10);**

**int selection;**

**do {**

**printf("===========STUDENT MANAGEMENT PROGRAM===========\n\n");**

**printf("1. Add a new student to the student list.\n");**

**printf("2. Modify information of an existing student in the student list based on ID.\n");**

**printf("3. Modify information of an existing student in the school database based on ID.\n");**

**printf("4. Display information of all student in the student list.\n");**

**printf("5. Display information of all student from the school database.\n");**

**printf("6. Find a student from the student list based on ID.\n");**

**printf("7. Find a student from the school database based on ID.\n");**

**printf("8. Delete a student from the student list based on ID.\n");**

**printf("9. Delete a student from the school database based on ID.\n");**

**printf("10. Add all students from the student list to the school database.\n");**

**printf("11. Retrieve all students from the school database and add them to the student list.\n");**

**puts("12. Display number of student in the student list.");**

**puts("13. Display number of student in the school database.");**

**puts("14. Average grades of all students in the list.");**

**puts("15. Average grades of all students in the school.");**

**puts("0. Exit.");**

**printf("\nEnter your selection: ");**

**while (!my\_scanf("%d", &selection)) {**

**printf("Please enter an integer.\n");**

**}**

**if (selection == 1) {**

**case\_1(list);**

**}**

**else if (selection == 2 || selection == 3) {**

**case\_2\_3(selection, list);**

**}**

**else if (selection == 4 || selection == 5) {**

**case\_4\_5(selection, list);**

**}**

**else if (selection == 6 || selection == 7) {**

**case\_6\_7(selection, list);**

**}**

**else if (selection == 8 || selection == 9) {**

**case\_8\_9(selection, list);**

**}**

**else if (selection == 10) {**

**case\_10(list);**

**}**

**else if (selection == 11) {**

**case\_11(list);**

**}**

**else if (selection == 12 || selection == 13) {**

**case\_12\_13(selection, list);**

**}**

**else if (selection == 14) {**

**avg\_grade\_in\_list(list);**

**}**

**else if (selection == 15) {**

**avg\_grade\_in\_database();**

**}**

**else if (selection == 0) {**

**int e = exit();**

**if (e == 1) {**

**break;**

**}**

**selection = e;**

**continue;**

**}**

**else {**

**printf("Invalid choice. Please enter a valid option\n\n");**

**pause();**

**}**

**} while (selection);**

**printf("Exiting program successfully.\n");**

**free\_student\_list(list);**

**return 0;**

**}**

function.h

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <stdbool.h>**

**#include <mysql/mysql.h>**

**#include "type.h"**

**#include "std\_string.h"**

**void initialize\_student\_list(student\_list\* list, int initial\_capacity);**

**void free\_student\_list(student\_list\* list);**

**void initialize\_student(student\*, int, const char\*, float, float, float, float, float, float, float, float, float);**

**void display\_infor\_of\_student(student);**

**// average grade for a student s**

**float avg\_grade(student\* s);**

**// average grade for all students in the student list**

**void avg\_grade2(student\_list\* list);**

**void avg\_grade\_in\_list(student\_list\* list);**

**// average grade for all students in the school database**

**void avg\_grade\_in\_database();**

**// add a student to the student list**

**int add\_student\_to\_student\_list(student\_list\* list, student new\_student);**

**// find student base on ID**

**void find(int id, student\_list\* list);**

**// modify student base on ID**

**int modify(int id, student\_list\*);**

**// delete student in the list base on ID**

**void deletion(int id, student\_list\* list);**

**void display\_all\_student\_in\_the\_list(student\_list\* list);**

**void display\_all\_student\_in\_the\_database();**

**// stop console**

**void pause();**

**// print number of student in list**

**void number\_of\_student(student\_list\* list);**

**// my safer scanf**

**int my\_scanf(const char\*, void\*);**

**void display(student\_list\* list);**

**int read\_from\_mysql(student\_list\* list, int list\_size);**

**int write\_to\_mysql(student\_list\* list);**

**void delete\_from\_mysql(int id);**

**int find\_from\_mysql(int id, student\_list\*);**

**int exit();**

**void case\_1(student\_list\* list);**

**void case\_2\_3(int decision, student\_list\* list);**

**void case\_4\_5(int decision, student\_list\* list);**

**void case\_6\_7(int decision, student\_list\* list);**

**void case\_8\_9(int decision, student\_list\* list);**

**void case\_10(student\_list\* list);**

**void case\_11(student\_list\* list);**

**void case\_12\_13(int decision, student\_list\* list);**

**int my\_scanf(const char\* format, void\* output) {**

**char buffer[100];**

**if (fgets(buffer, sizeof(buffer), stdin) == NULL) {**

**printf("Error occured when read from the key board.\n");**

**return 0;**

**}**

**if (sscanf(buffer, format, output) != 1) {**

**printf("Error occured when converting data.\n");**

**return 0;**

**}**

**return 1;**

**}**

**void initialize\_student\_list(student\_list\* list, int initial\_capacity) {**

**list->arr = (student\*)malloc(initial\_capacity \* sizeof(student));**

**list->capacity = initial\_capacity;**

**list->size = 0;**

**}**

**void add\_base\_on\_avg\_grade(student\_list\* list, student new\_student) {**

**list->arr[5] = new\_student;**

**int index = 0;**

**int n = list->size;**

**list->size++;**

**while (index < n) {**

**if (new\_student.avg\_grade < list->arr[index].avg\_grade) {**

**index++;**

**continue;**

**}**

**break;**

**}**

**if (index != n) {**

**for (int i = n - 1; i >= index; i--) {**

**list->arr[i + 1] = list->arr[i];**

**}**

**}**

**list->arr[index] = new\_student;**

**}**

**int add\_student\_to\_student\_list(student\_list\* list, student new\_student) {**

**if (list->size == list->capacity) {**

**list->capacity \*= 2;**

**list->arr = (student\*)realloc(list->arr, list->capacity \* sizeof(student));**

**}**

**student\_list\* temp\_list;**

**initialize\_student\_list(temp\_list, 1);**

**if (find\_from\_mysql(new\_student.id, temp\_list)) {**

**printf("The student with id %d is already exist in the school database.\n", new\_student.id);**

**return 0;**

**}**

**int k = list->size;**

**for (int i = 0; i < k; i++) {**

**if (new\_student.id == list->arr[i].id) {**

**printf("The student with id %d is already exist in the student list.\n", new\_student.id);**

**return 0;**

**}**

**}**

**add\_base\_on\_avg\_grade(list, new\_student);**

**return 1;**

**}**

**void free\_student\_list(student\_list\* list) {**

**free(list->arr);**

**list->arr = NULL;**

**free(list);**

**}**

**void initialize\_student(student\* s, int id, const char\* name, float math, float lit, float eng, float phys, float ch, float bio, float his, float geo, float cs) {**

**strcpy(s->full\_name, name);**

**s->id = id;**

**s->grades[0] = math;**

**s->grades[1] = lit;**

**s->grades[2] = eng;**

**s->grades[3] = phys;**

**s->grades[4] = ch;**

**s->grades[5] = bio;**

**s->grades[6] = his;**

**s->grades[7] = geo;**

**s->grades[8] = cs;**

**s->avg\_grade = avg\_grade(s);**

**}**

**void display\_infor\_of\_student(student s) {**

**printf("ID: %d\n", s.id);**

**printf("Name: %s\n", s.full\_name);**

**char subjects[9][17] = { "Math", "Literature", "English", "Physics", "Chemistry", "Biology", "History", "Geography", "Computer Science" };**

**printf("Subjects  ");**

**for (int i = 0; i < 9; i++) {**

**printf("%-12s", subjects[i]);**

**}**

**printf("  Average Grade");**

**printf("\nGrades    ");**

**for (int i = 0; i < 9; i++) {**

**printf("%-12.2f", s.grades[i]);**

**}**

**printf("      %.2f", s.avg\_grade);**

**printf("\n");**

**}**

**float avg\_grade(student\* s) {**

**float sum = 0;**

**int n = sizeof(s->grades) / sizeof(float);**

**for (int i = 0; i < n; i++) {**

**sum += s->grades[i];**

**}**

**float grade = sum / n;**

**return grade;**

**}**

**void avg\_grade2(student\_list\* list) {**

**student s;**

**float k;**

**float max = 0.0;**

**int max\_index;**

**float min = 10.0;**

**int min\_index;**

**int n = list->size;**

**int m = sizeof(s.grades) / sizeof(float);**

**for (int i = 0; i < m; i++) {**

**k = 0;**

**for (int j = 0; j < n; j++) {**

**k += list->arr[j].grades[i];**

**}**

**s.grades[i] = k / n;**

**if (s.grades[i] > max) {**

**max = s.grades[i];**

**max\_index = i;**

**}**

**if (s.grades[i] < min) {**

**min = s.grades[i];**

**min\_index = i;**

**}**

**}**

**k = 0;**

**for (int i = 0; i < n; i++) {**

**k += list->arr[i].avg\_grade;**

**}**

**s.avg\_grade = k / n;**

**puts(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**puts("|  Math  |  Lit.  |  Eng.  | Physic |  Che.  |  Bio.  |  His.  |  Geo.  |Comp. Sci.| Average |");**

**puts("|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|");**

**for (int j = 0; j < m - 1; j++) {**

**printf("|  %-5.2f ", s.grades[j]);**

**}**

**printf("|   %-5.2f  ", s.grades[m - 1]);**

**printf("|  %.2f   |", s.avg\_grade);**

**printf("\n");**

**puts("|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\n");**

**char subjects[9][17] = { "Math", "Literature", "English", "Physics", "Chemistry", "Biology", "History", "Geography", "Computer Science" };**

**printf("%s has the highest average grade, with a score of %.2f.\n", subjects[max\_index], max);**

**printf("%s has the lowest average grade, with a score of %.2f.\n", subjects[min\_index], min);**

**pause();**

**}**

**void avg\_grade\_in\_list(student\_list\* list) {**

**if (list->size == 0) {**

**puts("Students list is empty.");**

**pause();**

**return;**

**}**

**puts("Average grade of all students in the list: ");**

**avg\_grade2(list);**

**}**

**void avg\_grade\_in\_database() {**

**student\_list\* list = (student\_list\*)malloc(sizeof(student\_list));**

**initialize\_student\_list(list, 10);**

**read\_from\_mysql(list, 0);**

**if (list->size == 0) {**

**puts("Students list is empty.");**

**return;**

**}**

**puts("Average grade of all students in the school database: ");**

**avg\_grade2(list);**

**}**

**void find(int id, student\_list\* list) {**

**int n = list->size;**

**for (int i = 0; i < n; i++) {**

**if (list->arr[i].id = id) {**

**display\_infor\_of\_student(list->arr[i]);**

**return;**

**}**

**}**

**printf("There are no student with id %d in the student list.", id);**

**}**

**int modify(int id, student\_list\* list) {**

**int n = list->size;**

**for (int i = 0; i < n; i++) {**

**if (list->arr[i].id == id) {**

**char subjects[9][17] = { "Math", "Literature", "English", "Physics", "Chemistry", "Biology", "History", "Geography", "Computer Science" };**

**int num;**

**do {**

**printf("\nStudent information:\n");**

**display\_infor\_of\_student(list->arr[i]);**

**printf("\t\t----SELECT INFORMATION TO MODIFY----\n");**

**printf("1. ID\n2. Full name\n");**

**for (int j = 0; j < 9; j++) {**

**printf("%d. %s grade\n", j + 3, subjects[j]);**

**}**

**printf("0.Exit\n");**

**printf("Your selection is: ");**

**while (!my\_scanf("%d", &num)) {**

**printf("Please enter an integer.\n");**

**}**

**if (num == 1) {**

**printf("Enter new ID: ");**

**while (!my\_scanf("%d", &(list->arr[i].id))) {**

**printf("Please enter an integer.\n");**

**}**

**}**

**else if (num == 2) {**

**printf("Enter new full name: ");**

**fgets(list->arr[i].full\_name, 24, stdin);**

**fix\_string(list->arr[i].full\_name);**

**}**

**else if (num >= 3 && num <= 11) {**

**printf("Enter new grade for %s: ", subjects[num - 3]);**

**while (!my\_scanf("%f", &(list->arr[i].grades[num - 3]))) {**

**printf("Please enter an integer.\n");**

**}**

**list->arr[i].avg\_grade = avg\_grade(&(list->arr[i]));**

**}**

**else if (num == 0) {**

**printf("Exit successfully.");**

**pause();**

**break;**

**}**

**else {**

**printf("Invalid input. Enter again.");**

**continue;**

**}**

**} while (num);**

**return 1;**

**}**

**}**

**return 0;**

**}**

**void deletion(int id, student\_list\* list) {**

**int n = list->size;**

**for (int i = 0; i < n; i++) {**

**if (list->arr[i].id == id) {**

**if (i < n - 1) {**

**list->arr[i] = list->arr[n - 1];**

**}**

**list->size--;**

**printf("Remove student %d suceccfully.\n", id);**

**return;**

**}**

**}**

**printf("There are no student with id %d in the school.\n", id);**

**}**

**void case\_1(student\_list\* list) {**

**char full\_name[24];**

**int id;**

**float grade[9];**

**char subjects[9][17] = { "Math", "Literature", "English", "Physics", "Chemistry", "Biology", "History", "Geography", "Computer Science" };**

**char verify; // for verify if the entered grade is valid?**

**student s;**

**do {**

**printf("Enter ID: ");**

**while (!my\_scanf("%d", &id)) {**

**printf("Please enter an integer.\n");**

**}**

**printf("Enter full name: ");**

**fgets(full\_name, 24, stdin);**

**fix\_string(full\_name);**

**int n = sizeof(grade) / sizeof(float);**

**for (int i = 0; i < n; i++) {**

**printf("Enter grade for %s: ", subjects[i]);**

**while (!my\_scanf("%f", &grade[i])) {**

**printf("Please enter an integer.\n");**

**}**

**if (grade[i] > 10 || grade[i] < 0) {**

**printf("Grade must be greater than or equal 0 and less than or equal to 10. Enter again.\n");**

**i--;**

**}**

**}**

**initialize\_student(&s, id, full\_name, grade[0], grade[1], grade[2], grade[3], grade[4], grade[5], grade[6], grade[7], grade[8]);**

**printf("Student information:\n");**

**display\_infor\_of\_student(s);**

**do {**

**printf("\n\nVerify if the data is correct? (Y/N): ");**

**my\_scanf("%c", &verify);**

**if (verify != 'y' && verify != 'Y' && verify != 'n' && verify != 'N')**

**printf("\nInvalid command!");**

**} while (verify != 'y' && verify != 'Y' && verify != 'n' && verify != 'N');**

**} while (verify == 'N' || verify == 'n');**

**initialize\_student(&s, id, full\_name, grade[0], grade[1], grade[2], grade[3], grade[4], grade[5], grade[6], grade[7], grade[8]);**

**int x = add\_student\_to\_student\_list(list, s);**

**if (x)**

**printf("\nStudent with ID %d was added to the list.\n", s.id);**

**pause();**

**}**

**int read\_from\_mysql(student\_list\* list, int list\_size) {**

**MYSQL\* conn;**

**MYSQL\_RES\* result;**

**MYSQL\_ROW row;**

**conn = mysql\_init(NULL);**

**if (conn == NULL) {**

**fprintf(stderr, "mysql\_init() failed\n");**

**return 0;**

**}**

**if (mysql\_real\_connect(conn, "localhost", "Hoang", "Hoangdeptry\_05", "student", 0, NULL, 0) == NULL) {**

**fprintf(stderr, "mysql\_real\_connect() failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**if (mysql\_query(conn, "select student.id, fullname, math, literature, english, physic, chemistry, biology, history, geography, computer\_science, avg\_grade from student, grades where student.id = grades.id order by avg\_grade desc")) {**

**fprintf(stderr, "mysql\_query failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**result = mysql\_use\_result(conn);**

**int num\_fields = mysql\_num\_fields(result);**

**if (result == NULL) {**

**fprintf(stderr, "mysql\_use\_result() failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**student s;**

**char a[30];**

**while ((row = mysql\_fetch\_row(result)) != NULL) {**

**int i = 0;**

**while (i < num\_fields) {**

**strcpy(a, row[i]);**

**if (i == 0) {**

**sscanf(a, "%d", &s.id);**

**}**

**else if (i == 1) {**

**strcpy(s.full\_name, a);**

**}**

**else if (2 <= i && i <= 10) {**

**sscanf(a, "%f", &s.grades[i - 2]);**

**}**

**else {**

**sscanf(a, "%f", &s.avg\_grade);**

**}**

**i++;**

**}**

**if (list->size == list->capacity) {**

**list->capacity \*= 2;**

**list->arr = (student\*)realloc(list->arr, list->capacity \* sizeof(student));**

**}**

**int label = 1;**

**for (int i = 0; i < list\_size; i++) {**

**if (s.id == list->arr[i].id) {**

**label = 0;**

**break;**

**}**

**}**

**if (label == 1) {**

**list->arr[list->size] = s;**

**list->size++;**

**}**

**}**

**mysql\_close(conn);**

**return 1;**

**}**

**int count() {**

**MYSQL\* conn;**

**MYSQL\_RES\* result;**

**MYSQL\_ROW row;**

**conn = mysql\_init(NULL);**

**if (conn == NULL) {**

**fprintf(stderr, "mysql\_init() failed\n");**

**return 0;**

**}**

**if (mysql\_real\_connect(conn, "localhost", "Hoang", "Hoangdeptry\_05", "student", 0, NULL, 0) == NULL) {**

**fprintf(stderr, "mysql\_real\_connect() failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**if (mysql\_query(conn, "SELECT COUNT(\*) FROM student")) {**

**fprintf(stderr, "%s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**int number\_of\_record;**

**MYSQL\_RES\* res;**

**res = mysql\_use\_result(conn);**

**row = mysql\_fetch\_row(res);**

**number\_of\_record = atoi(row[0]);**

**mysql\_close(conn);**

**return number\_of\_record;**

**}**

**int find\_from\_mysql(int id, student\_list\* list) {**

**MYSQL\* conn;**

**MYSQL\_RES\* result;**

**MYSQL\_ROW row;**

**conn = mysql\_init(NULL);**

**if (conn == NULL) {**

**fprintf(stderr, "mysql\_init() failed\n");**

**return 0;**

**}**

**if (mysql\_real\_connect(conn, "localhost", "Hoang", "Hoangdeptry\_05", "student", 0, NULL, 0) == NULL) {**

**fprintf(stderr, "mysql\_real\_connect() failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**char\* query = (char\*)malloc(sizeof(char) \* 300);**

**strcpy(query, "select student.id, fullname, math, literature, english, physic, chemistry, biology, history, geography, computer\_science, avg\_grade from student, grades where student.id = grades.id and student.id = ");**

**char str[10];**

**sprintf(str, "%d", id);**

**strcat(query, str);**

**if (mysql\_query(conn, query)) {**

**fprintf(stderr, "mysql\_query failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**result = mysql\_use\_result(conn);**

**int num\_fields = mysql\_num\_fields(result);**

**student s;**

**char a[30];**

**if ((row = mysql\_fetch\_row(result)) != NULL) {**

**int i = 0;**

**while (i < num\_fields) {**

**strcpy(a, row[i]);**

**if (i == 0) {**

**sscanf(a, "%d", &s.id);**

**}**

**else if (i == 1) {**

**strcpy(s.full\_name, a);**

**}**

**else if (2 <= i && i <= 10) {**

**sscanf(a, "%f", &s.grades[i - 2]);**

**}**

**else {**

**sscanf(a, "%f", &s.avg\_grade);**

**}**

**i++;**

**}**

**list->arr[0] = s;**

**list->size++;**

**return 1;**

**}**

**return 0;**

**}**

**int write\_to\_mysql(student\_list\* list) {**

**if (list->size == 0) {**

**printf("There are no student in the student list.\n");**

**return 0;**

**}**

**MYSQL\* conn;**

**conn = mysql\_init(NULL);**

**if (conn == NULL) {**

**fprintf(stderr, "mysql\_init() failed\n");**

**return 0;**

**}**

**if (mysql\_real\_connect(conn, "localhost", "Hoang", "Hoangdeptry\_05", "student", 0, NULL, 0) == NULL) {**

**fprintf(stderr, "mysql\_real\_connect() failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return 0;**

**}**

**char\* query = (char\*)malloc(sizeof(char) \* 5000);**

**int num = list->size;**

**int num\_fields = 2;**

**strcpy(query, "insert ignore into student(id, fullname) value ");**

**for (int i = 0; i < num; i++) {**

**sprintf(query + strlen(query), "(%d, '%s')", list->arr[i].id, list->arr[i].full\_name);**

**if (i < num - 1) strcat(query, ", ");**

**}**

**if (mysql\_query(conn, query)) {**

**fprintf(stderr, "mysql\_query() failed: %s\n", mysql\_error(conn));**

**free(query);**

**mysql\_close(conn);**

**return 0;**

**}**

**num\_fields = 11;**

**strcpy(query, "insert ignore into grades(id, math, literature, english, physic, chemistry, biology, history, geography, computer\_science, avg\_grade) value ");**

**for (int i = 0; i < num; i++) {**

**sprintf(query + strlen(query), "(%d, %.2f, %.2f, %.2f, %.2f, %.2f, %.2f, %.2f, %.2f, %.2f, %.2f)", list->arr[i].id, list->arr[i].grades[0], list->arr[i].grades[1], list->arr[i].grades[2], list->arr[i].grades[3], list->arr[i].grades[4], list->arr[i].grades[5], list->arr[i].grades[6], list->arr[i].grades[7], list->arr[i].grades[8], list->arr[i].avg\_grade);**

**if (i < num - 1) strcat(query, ", ");**

**}**

**if (mysql\_query(conn, query)) {**

**fprintf(stderr, "mysql\_query() failed: %s\n", mysql\_error(conn));**

**free(query);**

**mysql\_close(conn);**

**return 0;**

**}**

**mysql\_close(conn);**

**free\_student\_list(list);**

**list = (student\_list\*)malloc(sizeof(student\_list));**

**initialize\_student\_list(list, 10);**

**return 1;**

**}**

**void delete\_from\_mysql(int id) {**

**MYSQL\* conn;**

**conn = mysql\_init(NULL);**

**if (conn == NULL) {**

**fprintf(stderr, "mysql\_init() failed\n");**

**return;**

**}**

**if (mysql\_real\_connect(conn, "localhost", "Hoang", "Hoangdeptry\_05", "student", 0, NULL, 0) == NULL) {**

**fprintf(stderr, "mysql\_real\_connect() failed: %s\n", mysql\_error(conn));**

**mysql\_close(conn);**

**return;**

**}**

**char query[100];**

**char str[100];**

**strcpy(query, "delete from student where id = ");**

**sprintf(str, "%d;", id);**

**strcat(query, str);**

**mysql\_query(conn, query);**

**strcpy(query, "delete from grades where id = ");**

**sprintf(str, "%d;", id);**

**strcat(query, str);**

**mysql\_query(conn, query);**

**mysql\_close(conn);**

**}**

**void display(student\_list\* list) {**

**int n = list->size;**

**int k = sizeof(list->arr[0].grades) / sizeof(float);**

**for (int i = 0; i < n; i++) {**

**printf("|  %-4d|%-10d| %-23s", i + 1, list->arr[i].id, list->arr[i].full\_name);**

**for (int j = 0; j < k - 1; j++) {**

**printf("|  %-5.2f ", list->arr[i].grades[j]);**

**}**

**printf("|   %-5.2f  ", list->arr[i].grades[k - 1]);**

**printf("|  %.2f  |", list->arr[i].avg\_grade);**

**printf("\n");**

**}**

**}**

**void display\_all\_student\_in\_the\_list(student\_list\* list) {**

**int n = list->size;**

**puts(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**puts("|      |         Identification            |                                         Grades                                            |");**

**puts("|S. No.|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|");**

**puts("|      |    ID    |       Full name        |  Math  |  Lit.  |  Eng.  | Physic |  Che.  |  Bio.  |  His.  |  Geo.  |Comp. Sci.|   Avg  |");**

**puts("|\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|");**

**display(list);**

**puts("|\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_|");**

**}**

**void display\_all\_student\_in\_the\_database() {**

**student\_list\* list = (student\_list\*)malloc(sizeof(student\_list));**

**initialize\_student\_list(list, 100);**

**read\_from\_mysql(list, 0);**

**if (list->size == 0) {**

**printf("Database empty!\n");**

**}**

**display\_all\_student\_in\_the\_list(list);**

**free\_student\_list(list);**

**}**

**void number\_of\_student(student\_list\* list) {**

**printf("Number of student in the student list: %d\n", list->size);**

**pause();**

**}**

**void pause() {**

**printf("Press Enter to continue ........");**

**char x = getchar();**

**if (x != '\n') {**

**my\_scanf("%c", &x);**

**}**

**}**

**void case\_2\_3(int selection, student\_list\* list) {**

**int id;**

**printf("\nEnter the ID of the student whose information needs to be modified: ");**

**while (!my\_scanf("%d", &id)) {**

**printf("Please enter an integer.\n");**

**}**

**if (selection == 2) {**

**if (!modify(id, list)) {**

**printf("There are no student with ID %d in the list.\n", id);**

**pause();**

**}**

**}**

**else {**

**student\_list\* temp\_list = (student\_list\*)malloc(sizeof(student\_list));**

**initialize\_student\_list(temp\_list, 1);**

**find\_from\_mysql(id, temp\_list);**

**if (!modify(id, temp\_list)) {**

**printf("There are no student with ID %d in the database.\n", id);**

**pause();**

**}**

**delete\_from\_mysql(id);**

**write\_to\_mysql(temp\_list);**

**free\_student\_list(temp\_list);**

**}**

**}**

**void case\_4\_5(int selection, student\_list\* list) {**

**if (selection == 4) {**

**if (list->size == 0) {**

**printf("Students list is empty.\n\n");**

**pause();**

**return;**

**}**

**printf("\n\n                                                ========LIST OF STUDENT=========\n\n");**

**display\_all\_student\_in\_the\_list(list);**

**}**

**else {**

**printf("\n\n                                                ========LIST OF STUDENT=========\n\n");**

**display\_all\_student\_in\_the\_database();**

**}**

**printf("\n");**

**pause();**

**}**

**void case\_6\_7(int selection, student\_list\* list) {**

**int id;**

**printf("\nEnter the ID of the student who you want to find: ");**

**while (!my\_scanf("%d", &id)) {**

**printf("Please enter an integer.\n");**

**}**

**student s;**

**int in = 0;**

**if (selection == 6) {**

**find(id, list);**

**}**

**else {**

**student\_list\* temp\_list = (student\_list\*)malloc(sizeof(student\_list));**

**initialize\_student\_list(temp\_list, 1);**

**if (find\_from\_mysql(id, temp\_list)) {**

**display\_all\_student\_in\_the\_list(temp\_list);**

**}**

**else {**

**printf("\nThere are no student with ID %d in the school database.\n", id);**

**}**

**free\_student\_list(temp\_list);**

**}**

**pause();**

**}**

**void case\_8\_9(int selection, student\_list\* list) {**

**int id;**

**printf("\nEnter the ID of the student who you want to remove: ");**

**while (!my\_scanf("%d", &id)) {**

**printf("Please enter an integer.\n");**

**}**

**if (selection == 8) {**

**deletion(id, list);**

**}**

**else {**

**delete\_from\_mysql(id);**

**}**

**pause();**

**}**

**void case\_10(student\_list\* list) {**

**if (write\_to\_mysql(list) == 1) {**

**printf("Write data to database successfuly.\n");**

**}**

**else {**

**printf("Failed to write data to the database. An error has occured\n");**

**}**

**pause();**

**}**

**void case\_11(student\_list\* list) {**

**if (read\_from\_mysql(list, list->size) == 1) {**

**printf("Retrieve data from database successfuly.\n");**

**}**

**else {**

**printf("Failed to retrieve data from the database. An error has occured");**

**}**

**pause();**

**}**

**void case\_12\_13(int selection, student\_list\* list) {**

**if (selection == 12) {**

**number\_of\_student(list);**

**}**

**else {**

**printf("Number of student in the school database: %d\n", count());**

**pause();**

**}**

**}**

**int exit() {**

**char decision;**

**do {**

**printf("Are you sure that you want to exit the program? (Y/N)  ");**

**my\_scanf("%c", &decision);**

**if (decision == 'y' || decision == 'Y' || decision == 'n' || decision == 'N') {**

**break;**

**}**

**else {**

**printf("Invalid choice. Please enter a valid option\n");**

**continue;**

**}**

**} while (1);**

**if (decision == 'y' || decision == 'Y') {**

**return 1;**

**}**

**else {**

**return -1;**

**}**

**}**

std\_string.h

**#include <stdio.h>**

**#include<string.h>**

**#include <stdlib.h>**

**bool is\_white\_space(char x) {**

**return x == ' ' || x == '\n' || x == '\t' || x == '\r';**

**}**

**int count\_word(char\* str) {**

**int count = 0;**

**bool is\_in\_word = false;**

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

**if (is\_white\_space(str[i])) {**

**is\_in\_word = false;**

**}**

**else {**

**if (is\_in\_word == false) {**

**count++;**

**is\_in\_word = true;**

**}**

**}**

**}**

**return count;**

**}**

**void standardized\_word(char\* str) {**

**int n = strlen(str);**

**if (str[0] >= 97 && str[0] <= 122) {**

**str[0] -= 32;**

**}**

**for (int i = 1; i < n; i++) {**

**if (str[i] <= 90 && str[i] >= 65) {**

**str[i] += 32;**

**}**

**}**

**}**

**void remove\_first\_word(char\* str) {**

**bool is\_word = false;**

**int length = strlen(str);**

**int j = 0;**

**while (is\_white\_space(str[j]) && j < length) {**

**j++;**

**}**

**while (!is\_white\_space(str[j]) && j < length) {**

**str[j] = ' ';**

**j++;**

**}**

**}**

**void standardized(char\* str) {**

**int l = strlen(str);**

**int count = count\_word(str);**

**char\*\* std\_str = (char\*\*)malloc(count \* sizeof(char\*));**

**for (int i = 0; i < count; i++) {**

**std\_str[i] = (char\*)malloc(10 \* sizeof(char));**

**}**

**for (int i = 0; i < count; i++) {**

**sscanf(str, "%10s", std\_str[i]);**

**remove\_first\_word(str);**

**standardized\_word(std\_str[i]);**

**}**

**int j = 0;**

**for (int i = 0; i < count; i++) {**

**int l = strlen(std\_str[i]);**

**if (j > 0)str[j] = ' ';**

**for (int k = 0; k < l; k++) {**

**str[j] = std\_str[i][k];**

**j++;**

**}**

**j++;**

**}**

**str[j - 1] = '\0';**

**}**

**void fix\_string(char\* str) {**

**int length = strlen(str);**

**standardized(str);**

**if (length >= 2) {**

**if (str[length - 1] == '\n' && str[length - 2] == '\r') {**

**str[length - 2] = '\0';**

**}**

**else if (str[length - 1] == '\n' && str[length - 2] != '\r') {**

**str[length - 1] = '\0';**

**}**

**}**

**}**

type.h

**#include <stdio.h>**

**typedef struct student {**

**char full\_name[24];**

**int id;**

**float grades[9]; // 0.math, 1.literature, 2.english, 3.physic, 4.chemistry, 5.biology, 6.history, 7.geography, 8.computer science**

**float avg\_grade;**

**} student;**

**typedef struct student\_list {**

**student\* arr;**

**int capacity; // capacity of the array**

**int size; // number of student**

**} student\_list;**