**Vấn đề 2:** **Quản lý điểm sinh viên**

Điểm học phần Kỹ thuật lập trình của sinh viên được quản lý bằng một mảng cấu trúc (mỗi phần tử lưu thông tin và điểm thi của một sinh viên).

**Yêu cầu:**

1. **Định nghĩa** kiểu cấu trúc SINHVIEN quản lý điểm sinh viên gồm: Họ tên, Mã sinh viên, điểm thường kỳ (TK), điểm giữa kỳ (GK), điểm cuối kỳ (CK), điểm thực hành (TH) và điểm trung bình (DTB).

typedef struct SINHVIEN {

char name[30];

char MaSV[10];

float TK, GK, CK, TH, DTB;

}sv;

1. **Đọc dữ liệu** từ bàn phím (hoặc từ file) vào Mảng cấu trúc của lớp sinh viên (n sinh viên)

SINHVIEN A[MAX]; ***// #define MAX 100***

void input(SINHVIEN A[], int n) {

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

printf("---------Thong tin cua sinh vien thu %d-----------\n", i+1);

fflush(stdin);

printf("Ho va ten: "); gets(A[i].name);

printf("Ma Sinh Vien: "); gets(A[i].MaSV);

printf("Diem thuong ki: "); scanf("%f", &A[i].TK);

printf("Diem giua ki: "); scanf("%f", &A[i].GK);

printf("Diem cuoi ki: "); scanf("%f", &A[i].CK);

printf("Diem thuc hanh: "); scanf("%f", &A[i].TH);

diemTB(A[i]);

}

}

1. **Tính** điểm trung bình (DTB) của mỗi sinh viên theo công thức:

float diemTB(SINHVIEN &sv) {

float dtb = (((2 \* sv.TK + 3 \* sv.CK + 5 \* sv.CK) / 10) \* 2 + sv.TH) / 3;

return dtb;

}

1. **Hiển thị** lên danh sách sinh viên phải học lại (có DTB <4.0) theo khuôn dạng:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ TÊN | MÃ SINH VIÊN | ĐIỂM TB |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

void HocLai(SINHVIEN A[], int n) {

int stt = 0;

printf("\n---------Danh sach sinh vien phai hoc lai-------------\n");

printf("STT\tHO TEN\tMA SINH VIEN\tDIEM TB\n");

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

if (A[i].DTB < 4.0) {

printf("%d\t%s\t%s\t%.2f\n", i+1, A[i].name, A[i].MaSV, A[i].DTB);

stt = 1;

}

}

if (stt == 0) {

printf("Khong co sinh vien phai hoc lai!");

}

}

1. **Tìm** sinh viên có Mã sinh viên X được nhập từ bàn phím. Hiển thị lên màn hình thông tin kết quả học tập của sinh viên đó, hoặc thông báo “Khong tìm thay” nếu không tồn tại sinh viên có mã X.

Ho ten: ...............................

Ma sinh vien: .........................

Diem thuong ky: .......................

Diem giua ky: .........................

Diem cuoi ky: .........................

Diem thuc hanh: .......................

Diem trung binh: ......................

void timkiemSV(SINHVIEN A[], int n) {

int stt = 0;

char x[30];

printf("\nNhap MA SINH VIEN cua sinh vien can tim kiem: ");

fflush(stdin);

gets(x);

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

if (strcmp(A[i].MaSV, x) == 0) {

printf("Ho ten: %s\n", A[i].name);

printf("Ma sinh vien: %s\n", A[i].MaSV);

printf("Diem thuong ki: %.2f\n", A[i].TK);

printf("Diem giua ki: %.2f\n", A[i].GK);

printf("Diem cuoi ki: %.2f\n", A[i].CK);

printf("Diem thuc hanh: %.2f\n", A[i].TH);

printf("Diem trung binh: %.2f\n", A[i].DTB);

stt = 1;

}

}

if (stt == 0) {

printf("\nKhong tim thay!\n");

}

}

1. **Sắp xếp** lại dữ liệu để được danh sách theo thứ tự giảm dần của DTB

void sapxep(SINHVIEN A[], int n) {

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

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

if (A[i].DTB < A[j].DTB) {

SINHVIEN tmp = A[i];

A[i] = A[j];

A[j] = tmp;

}

}

}

printf("\n---------Danh sach sinh vien sap xep theo thu tu DTB giam dan-------------\n");

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

printf("Ho ten: %s\n", A[i].name);

printf("Ma sinh vien: %s\n", A[i].MaSV);

printf("Diem thuong ki: %.2f\n", A[i].TK);

printf("Diem giua ki: %.2f\n", A[i].GK);

printf("Diem cuoi ki: %.2f\n", A[i].CK);

printf("Diem thuc hanh: %.2f\n", A[i].TH);

printf("Diem trung binh: %.2f\n\n", A[i].DTB);

}

}

1. **Hiển thị** lên màn hình DANH SÁCH ĐIỂM THI KTLT theo khuôn dạng sau:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STT | HO TEN | MA SV | TK | GK | CK | TH | DTB | GHI CHU |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Cột ghi chú có nội dung “**Hoc lai**” nếu DTB < 4 hoặc để trống trong trường hợp ngược lại.

void output(SINHVIEN A[], int n) {

printf("\n-------------Danh sach DIEM THI KTLT----------------\n");

printf("STT\tHO TEN\t\t\t\tMA SV\t\tTK\tGK\tCK\tTH\tDTB\tGHI CHU\n");

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

if (A[i].DTB < 4.0) {

printf("%d\t%s\t\t\t%s\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\tHoc Lai\n", i+1, A[i].name, A[i].MaSV, A[i].TK, A[i].GK, A[i].CK, A[i].TH, A[i].DTB);

}

else {

printf("%d\t%s\t\t%s\t%.1f\t%.1f\t%.1f\t%.1f\t%.1f\t \n", i+1, A[i].name, A[i].MaSV, A[i].TK, A[i].GK, A[i].CK, A[i].TH, A[i].DTB);

}

}

printf("\n");

}