

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct aluno {
    char nome[20];
    char mat[10];
    float notas[5];
    int visto;
    int faltas;
};

struct turma {
    int qtd;
    char prof[20];
    char ID[5];
    struct aluno *aluno;
};

int main() {
    int cont1 = 0, cont2, cont3, cont4, cont5, opcao;
    char removeNome[20], removeMatricula[10], aprovadosID[5],
reprovadosID[5];
    struct turma *turma = calloc(0, sizeof(struct turma));
    struct aluno atual; // para filtro
    if (turma == NULL)
        exit(1);
    switch (scanf("%d", &opcao, (opcao <= 5 && opcao > 0))) {
    case 1:
        turma = realloc(turma, ++cont1);
        if (turma == NULL)
            exit(1);
        cont1--;
        printf("qtd de alunos\n");
        scanf("%d", &turma[cont1].qtd); // rever isso
        turma[cont1].aluno = // essa linha ta bugada, provavelmente erro
de sintaxe
                                // para alocação inicial
        (struct aluno *)calloc(turma[cont1].qtd, sizeof(struct
aluno));
        printf("nome do professor\n");
        scanf("%*c%[^\\n]", turma[cont1].prof);
        printf("ID\n");
        scanf("%*c%[^\\n]", turma[cont1].ID);
        printf("Lista de alunos\n");
        for (cont2 = 0; cont2 <= turma[cont1].qtd; cont2++) {
            scanf("%*c%[^\\n]",
                turma[cont1]
                    .aluno[cont2]
                    .nome); // aqui da segfault provavelmente devido a
linha 35-36
            scanf("%*c%[^\\n]", turma[cont1].aluno[cont2].mat);
            for (cont3 = 0; cont3 <= 4; cont3++)
                scanf("%f", &turma[cont1].aluno[cont2].notas[cont3]);
            for (cont3 = 0; cont3 <= 3; cont3++)
                turma[cont1].aluno[cont2].notas[cont3 + 1] +=

```

```

        turma[cont1].aluno[cont2].notas
            [cont3]; // pega a soma das notas, note que fica em
notas[4];
        scanf("%d", &turma[cont1].aluno[cont2].faltas);
        for (cont3 = 0; cont3 <= turma[cont1].qtd; cont3++)
            turma[cont1].aluno[cont2].visto = 0; // para filtro na ordem
alfabetica
    }
    break;
case 2:
    printf("nome do aluno e matricula\n");
    scanf("%*c%[\n]", removeNome);
    scanf("%*c%[\n]", removeMatricula);
    for (cont4 = 0; cont4 <= cont1; cont4++) // anda as turmas
        for (cont5 = 0; cont5 <= turma[cont4].qtd;
            cont5++) // anda alunos por turma
            if (!strcmp(turma[cont4].aluno[cont5].nome, removeNome) &&
                !strcmp(turma[cont4].aluno[cont5].mat, removeMatricula)) {
                turma[cont4].aluno[cont5] =
                    turma[cont4]
                        .aluno[turma[cont4].qtd]; // pega o cara do final e
joga para
                                                    // o cara que sera
removido e como
                                                    // ele foi repetido, tira
ele com um
                                                    // realloc

                turma[cont4].aluno =
                    realloc(turma[cont4].aluno, turma[cont4].qtd - 1);
                if (turma[cont4].aluno == NULL)
                    exit(1);
                turma[cont4].qtd--;
            }
    break;
case 3:
    printf("ID da turma\n");
    scanf("%*c%[\n]%*c", aprovadosID);
    for (cont4 = 0; cont4 <= cont1; cont4++) { // anda as turmas
        if (!strcmp(turma[cont4].ID, aprovadosID))
            for (cont5 = 0; cont5 <= turma[cont4].qtd;
                cont5++) // anda alunos por turma
                if ((turma[cont4].aluno[cont5].notas[4]) / 5 >= 7.0)
                    printf("Aprovado: %s\n", turma[cont4].aluno[cont5].nome);
        cont4 = cont1 + 1;
    }
    break;
case 4:
    printf("ID da turma\n");
    scanf("%*c%[\n]", reprovadosID);
    for (cont4 = 0; cont4 <= cont1; cont4++) { // anda as turmas
        if (!strcmp(turma[cont4].ID, reprovadosID))
            for (cont5 = 0; cont5 <= turma[cont4].qtd;
                cont5++) // anda alunos por turma
                if ((turma[cont4].aluno[cont5].notas[4]) / 5 < 3.0)
                    printf("Reprovado: %s\n", turma[cont4].aluno[cont5].nome);
        cont4 = cont1 + 1;
    }
}

```

```

    break;
case 5:
    for (cont2 = 0; cont2 <= cont1; cont2++) {
        printf("%s\n", turma[cont2].prof);
        strcpy(atual.nome, turma[cont2].aluno[0].nome);
        cont4 = 0;
        for (cont3 = 0; cont3 <= turma[cont2].qtd;
            cont3++) // checa a turma inteira
            while (cont4 <= turma[cont2].qtd) {
                if ((strcmp(turma[cont2].aluno[cont4].nome, atual.nome) < 0)
                &&
                    turma[cont2].aluno[cont4].visto == 0)
                    strcpy(atual.nome, turma[cont2].aluno[cont4].nome);
                cont4++;
            }
        printf("%s\n", atual.nome);
        turma[cont2].aluno[cont4].visto = 1;
    }
    break;
}
if (turma == NULL)
    exit(1);
for (cont2 = 0; cont2 <= cont1; cont2++) {
    if (turma[cont2].aluno == NULL)
        exit(1);
    free(turma[cont2].aluno);
}
if (turma == NULL)
    exit(1);
free(turma);
return 0;
}

```