

Nome: _____

1) (3.4 pontos) Considere a struct a seguir:

```
typedef struct venda{int dia;int mês;int ano;int quantidade;} tvenda;
```

Faça um programa que leia um arquivo binário “vendas.bin”, e gere um arquivo texto “totvendas.txt” com um relatório de vendas por mês, ordenada em ordem decrescente de total de vendas. Exemplo de relatório:

Mês	Vendas
7	1000000
5	900000
12	880000

2) (3.3 pontos) Faça uma função que receba uma matriz M[4][4] de inteiros iguais a 0 ou 1, e retorne um inteiro de 16 bits sem sinal em que cada bit contém o valor correspondente a uma posição da matriz. O bit 15 deve conter o valor da posição m[0][0] e o bit 0 deve conter o valor da posição [3][3]. Ex: Se a matriz M contém:

0	1	0	1
1	1	1	1
0	1	0	1
0	0	0	0

A função deve retornar o valor inteiro correspondente ao binário 0101 1111 0101 0000.

3) (3.3 pontos) Fazer uma função escrita na linguagem de programação C que mostre todos os nomes dos alunos armazenados no arquivo ‘alunos.dat’ em ordem contrária, ou seja, do último aluno gravado até o primeiro. Abaixo exemplo de um programa que armazena alguns alunos nesse arquivo e algumas funções que podem ser usadas. O programa deve funcionar para qualquer quantidade de registros e não para apenas os desse exemplo.

fseek(arq, 0, SEEK_SET) // posiciona na posição 0 (deslocamento em bytes) a partir do início do arquivo (SEEK_SET). Outras opções SEEK_END (fim do arquivo), SEEK_CUR (posição corrente).

ftell(arq) // Retorna a posição corrente no arquivo(em bytes) a partir da posição 0 (primeira posição do arquivo)

fread(&variavel, sizeof(variavel), 1, arq) lê para a variável a partir da posição corrente do arquivo. Retorna quantos foram lidos, nesse caso 0 se não foi lido, ou 1 se foi lido.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct st_aluno {
    int codaluno;
    char nomealuno[41];
    float notas[3];
    int faltas;
} TIPOALUNO;

void cria_arq_alunos() {
    FILE *arq;
    TIPOALUNO aluno;
    arq=fopen("alunos.dat","wb");
    if(arq==NULL) {
        printf("Problema ao criar alunos.dat\n");
        exit(2);
    }
    aluno=(TIPOALUNO) { 2,"pedro", {8.6,9.2,7.5}, 0}; fwrite(&aluno,sizeof(aluno),1,arq);
    aluno=(TIPOALUNO) { 3,"ana", {9.1,9.0,9.4}, 2}; fwrite(&aluno,sizeof(aluno),1,arq);
    aluno=(TIPOALUNO) { 5,"roberto",{6.0,6.2,6.6}, 3}; fwrite(&aluno,sizeof(aluno),1,arq);
    aluno=(TIPOALUNO) { 8,"maria", {7.9,6.9,8.9}, 1}; fwrite(&aluno,sizeof(aluno),1,arq);
    fclose(arq);
}

int main() {
    cria_arq_alunos();
    return 0;
}
```