Nome: _____ USP: _____

Laboratório de Introdução à Ciência da Computação

1. (*Aquecimento*) Escreva uma função que recebe dois valores inteiros como parâmetro e troca esses valores. Por exemplo, caso o primeiro parâmetro seja 3 e o segundo 9, após a execução da função o primeiro parâmetro deve ser 9 e o segundo 3. Implemente também a função main para testar a sua função “troca”. Detalhe: as duas variáveis passadas como parâmetro devem ser declaradas na função main.

2. O arquivo notas.dat contém a relação das notas dos trabalhos dos alunos de ICC. Ele foi gravado usando a seguinte estrutura:

```
struct reg_notas {  
    int nusp;  
    char nome[40];  
    double nota_grupo;  
    double nota_individual;  
    double media;  
    int proximo;  
}
```

Os campos desse registro contêm: o número USP do aluno, seu nome, a nota dada para o trabalho, a nota individual dada ao aluno e a média. O último campo indica qual é o próximo registro da sequência. No início do arquivo há um registro com dois números inteiros que indicam quantos registros foram gravados e qual é o primeiro registro da sequência. Por exemplo:

6	4				
					5
					1
					6
					2
					3
					-1

Esse arquivo mostra que existem 6 registros e o primeiro a ser lido é o registro 4. Depois do 4, o 2, o 1, o 5, o 3 e o 6. Faça um programa para descobrir as notas dos trabalhos. Esse programa deve ter as seguintes restrições:

- leia o número de registros que estão no arquivo e aloque de forma dinâmica um array para colocar cada um dos registros;
- leia o arquivo na ordem mostrada pelos próprios registros, conforme mostrado no exemplo acima. Use a função fseek para isso;
- escreva uma função le_registro que recebe como parâmetro o número do registro a ser lido, lê o registro do arquivo e o retorna para quem chamou;
- a função le_registro deve alocar de forma dinâmica a memória onde vão ser lidos os registros.

3. Faça um programa que leia um valor n e crie *dinamicamente* um array de n elementos e passe esse array para uma função que vai ler cada um dos seus elementos digitados pelo usuário. Depois, na função `main`, o array preenchido deve ser exibido. Adicionalmente, antes de finalizar o programa, deve-se liberar a área de memória alocada.

4. Escreva um programa que leia uma matriz $n \times n$ (sendo que o usuário deve digitar n) e armazene em um array os elementos da diagonal principal. Além disso, apresente também o menor e o maior elemento desse array. Tanto a matriz quanto o array devem ser *dinamicamente* alocados. Assim, é também deve-se criar uma função para liberar a área de memória alocada para esses dois elementos.

5. Criar um tipo abstrato de dados que represente um livro, contendo título, data de lançamento, edição, número de páginas e editora. Crie uma variável que é um ponteiro para este tipo de dados recém-criado (no programa principal). Em seguida, crie uma função que receba este ponteiro e preencha os dados da estrutura e uma função que também receba este ponteiro e imprima os valores de cada campo da estrutura. Finalmente, faça a chamada a estas funções na função principal (`main`).