Ficheiros de texto

NOTA: para testar os programas que serão implementados, usar os ficheiros de texto com os mesmos nomes que se encontra na página web da disciplina.

1. Implemente um programa em C que, sem usar arrays,

- **a)** determine a **soma** dos números **negativos** e a **soma** dos números **positivos** contidos no ficheiro "Inteiros.txt" (com números inteiros),
- **b)** guarde no ficheiro "Saida1.txt" os resultados obtidos.

2. Implemente um programa que, sem usar arrays,

- **a)** determine o **maior** número e o **menor** número contidos no ficheiro "Inteiros.txt",
- **b)** guarde no ficheiro "Saida1.txt" os resultados obtidos.

3. Implemente um programa em C que, sem usar arrays, realize as seguintes ações:

- **a)** determine o **maior** número contido no ficheiro "InteirosPositivos.txt",
- **b)** guarde no ficheiro "Saida2.txt" todos os **divisores** do **número** obtido em **b)** contidos no ficheiro "InteirosPositivos.txt".

4. Implemente um programa em C que, sem usar arrays, realize as seguintes ações:

- **a)** construa **um array** com os **números** contidos no ficheiro "InteirosPositivos.txt",
- **b)** determine o **valor médio** dos números contidos em "InteirosPositivos.txt" (é o inteiro mais próximo da média dos números contidos no ficheiro e que pode não estar no ficheiro),
- **c)** guarde no ficheiro "Saida21.txt" todos os números contidos no ficheiro "InteirosPositivos.txt" que são menores do que o valor médio obtido em b) e no ficheiro "Saida22.txt" os restantes números.

5. Implemente um programa em C que, sem usar arrays, realize as seguintes ações:

- **a)** peça ao utilizador e insira um valor inteiro **K** no intervalo [2000, 10000],
- **b)** determine a **quantidade** e a **soma** dos números que são **divisores de K**, que estão contidos no ficheiro "InteirosPositivos.txt",
- **c)** guarde no ficheiro "Saida2.txt" os resultados obtidos.

6. Implemente um programa em C que, sem usar arrays, realize as seguintes ações:

- **a)** construa um **array** com os números **positivos** contidos no ficheiro "Inteiros.txt",
- **b)** determine o **menor** número **positivo** contido no ficheiro "Inteiros.txt",
- **c)** guarde no ficheiro "Saida1.txt" todos os **múltiplos** do **menor** obtido em b) até 5000.

7. Escreva um programa que, sem usar arrays, peça ao utilizador um número inteiro positivo não nulo (> 0) e guarde os múltiplos desse valor que estão no ficheiro de texto "Multiplos.txt" no ficheiro "Saida.txt".

8. Implemente um programa que realize as seguintes ações:

a) determine o **segundo maior** número contido no ficheiro "InteirosPositivos.txt",

b) guarde no ficheiro "Saida2.txt" todos os **divisores** do **número** obtido em **b)** contidos no ficheiro "InteirosPositivos.txt".

9. Considere o ficheiro de texto "Dados1.txt" que deverá conter uma pauta com a seguinte informação (em 3 linhas seguidas): número do aluno, classificação da frequência, classificação do exame. Escreva um programa que crie outro ficheiro, contendo o nome do aluno e a classificação final, correspondente à melhor das duas notas, dos alunos aprovados (alunos com classificação final igual ou superior a 10).

10. As notas dos alunos da disciplina de Algoritmos estão guardadas no ficheiro de texto "Dados2.txt". Cada linha deste ficheiro contém a informação relativa a um aluno, pela seguinte ordem: número e nota final obtida.

Considere a seguinte estrutura:

```
typedef struct {
        int    numAluno;
        int    notaFinal;
} ALUNO;
```

Implementar um programa em C para

1) ler um **array 1D** do tipo **ALUNO** a partir dos dados do ficheiro
2) mostrar o **array 1D**
3) determinar e mostrar a **média** obtida na disciplina pelos alunos
4) remover os alunos reprovados (nota final inferior a 10)
5) mostrar o **array 1D** atualizado (só com aprovados)
6) determinar e mostrar a **média** obtida na disciplina pelos alunos aprovados

Para tal, construir e usar as seguintes funções:

a) construir um **array 1D** do tipo **ALUNO** com os dados do ficheiro, usando o cabeçalho:

ALUNO *lerArray1DAluno (**int** *N)

b) calcular a soma das notas obtidas pelos alunos, usando o cabeçalho:

int somaNotasAluno (**ALUNO** *V, **int** N)

c) remover todos os alunos com valor no campo **notaFinal** menor que um valor **K**, usando o cabeçalho:

ALUNO *removerElementos (**ALUNO** *V, **int** *N, **int** K)

d) escrever/mostrar no monitor um **array 1D** do tipo **ALUNO**, usando o cabeçalho:

void escreverArray1DAluno (**ALUNO** *X, **int** N)

11. As notas dos alunos da disciplina de Programação estão guardadas no ficheiro de texto "Dados2.txt". Cada linha deste ficheiro contém a seguinte informação: número de aluno e nota final obtida (por esta ordem). Considere a estrutura ALUNO definida no exercício anterior.

Implementar um programa em C para determinar a nota média obtida na disciplina. Para tal, construir e usar as seguintes funções:

a) função para ler do ficheiro um array de inteiros com as notas dos alunos.

void lerArray1DInteiros (int V[], int *N);

b) função para calcular a soma das notas obtidas pelos alunos (usar o array).

int somaNotas (int V[], int N);

12. As notas dos alunos da disciplina de Programação estão guardadas no ficheiro de texto "Dados2.txt". Cada linha do ficheiro contém a seguinte informação: número de aluno e nota final obtida (por esta ordem). Implementar um programa em C para determinar a nota média considerando os alunos aprovados. Para tal, construir e usar as seguintes funções:

a) função para ler do ficheiro um array de inteiros com as notas obtidas pelos alunos.

void lerArray1DInteiros (int V[], int *N);

b) função para calcular a soma das notas dos alunos aprovados (usar o array).

int somaNotasAprovados (int V[], int N);

13. As notas dos alunos da disciplina de Programação estão guardadas no ficheiro de texto "Dados3.txt". Cada linha deste ficheiro contém a seguinte informação: número de aluno, nota dos trabalhos práticos (TP) e nota do teste escrito (T) (por esta ordem). Implementar um programa em C para determinar o número de alunos aprovados na disciplina. Para tal, construir e usar as seguintes funções:

a) função para ler do ficheiro um array de inteiros com as notas finais (TP+T) dos alunos.

void lerArray1DInteiros (int V[], int *N);

b) função para calcular o número de alunos aprovados (usar o array).

int calcularAprovados (int V[], int N);