

Recursividade

- 1.** Implementar duas funções recursivas para calcular o valor de 2^N , com $N \geq 0$, uma usando a forma normal e a outra a terminal.
- 2.** Implementar duas funções recursivas para calcular o número de caracteres que contém uma frase, uma usando a forma normal e a outra a terminal, tendo em conta que os espaços em branco são caracteres a contabilizar e que a frase termina com '\n'.
- 3.** Implementar duas funções recursivas para calcular o número de dígitos que contém um número inteiro positivo N , uma usando a forma normal e a outra a terminal.
- 4.** Implementar duas funções recursivas para calcular o valor de $\text{Fib}(N)$, com $N \geq 0$, uma usando a forma normal e a outra a terminal, tendo em conta que:
$$\text{Fib}(0) = 1; \text{Fib}(1) = 1; \text{Fib}(N) = \text{Fib}(N-1) + \text{Fib}(N-2).$$
- 5.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para calcular o máximo divisor comum de dois números inteiros positivos, usando o algoritmo de Euclides, que se enuncia da seguinte forma:
$$\text{MDC}(a, 0) = 1; \text{MDC}(a, b) = \text{MDC}(b, a \% b).$$
- 6.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para testar se um dado número inteiro N é primo ou não. Um inteiro $N > 1$ é primo se e só se é apenas divisível por ele próprio e por 1.
- 7.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar o maior elemento de um vector de inteiros (usar o ficheiro "Inteiros.txt" que se encontra na página web da disciplina).
- 8.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar o índice/posição do maior elemento de um vector de inteiros (usar o ficheiro "Inteiros.txt" que se encontra na página web da disciplina).
- 9.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar a soma dos elementos de um vector de reais positivos (usar o ficheiro "ReaisPositivos.txt" que se encontra na página web da disciplina).
- 10.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar a soma dos elementos pares de um vector de inteiros positivos (usar o ficheiro "InteirosPositivos.txt" que se encontra na página web da disciplina).
- 11.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar a média dos elementos de um vector de reais positivos (usar o ficheiro "ReaisPositivos.txt" que se encontra na página web da disciplina).

- 12.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar o número de elementos pares de um vector de inteiros positivos (usar o ficheiro "InteirosPositivos.txt" que se encontra na página web da disciplina).
- 13.** Implementar duas funções recursivas, uma na sua forma normal e a outra na terminal, para determinar o número de elementos positivos de um vector de inteiros (usar o ficheiro "Inteiros.txt" que se encontra na página web da disciplina).
- 14.** Podemos representar um polinómio P de grau N por um vector p de reais em que uma célula de índice i representa o coeficiente associado à potência de grau i . Escreva um programa que peça ao utilizador o valor de N (com a restrição que $N \geq 0$), inicialize p e que para um dado x , calcule $P(x)$ usando o método de Horner de forma recursiva, i.e.
- $$P_N(x) = (\dots ((p_N x + p_{N-1})x + p_{N-2})x + \dots + p_1)x + p_0$$
- 15.** Modifique a implementação do método de Horner de tal forma que a leitura do vector de coeficientes do polinómio seja implementada sob a forma de uma função recursiva.
- 16.** Seja $\text{par}(n)$ a função que devolve 1 se n for par e 0 no caso contrário. E seja $\text{impar}(n)$ a função que devolve 1 se n for ímpar e 0 no caso contrário.
- a)** Defina as funções par e impar de forma mutuamente recursiva.
 - b)** Apresente uma definição sem recursividade mútua.
- 17.** Para $(n,p) \in \mathbb{N} \times \mathbb{N}$ com $0 \leq p \leq n$ a função $C(n,p)$ pode-se calcular usando a seguinte expressão:
- $$C(n,0) = 1; C(n,n) = 1; C(n,p) = C(n-1,p) + C(n-1,p-1), \text{ se } 0 < p < n.$$
- Implementar uma função recursiva para determinar o valor de $C(n,p)$.
- 18.** 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 e nota final obtida (por esta ordem). Implementar um programa em C para determinar a nota média obtida na disciplina e a nota média obtida na disciplina pelos alunos aprovados. Para tal, construir e usar as seguintes funções:
- a)** função para ler do ficheiro um vector de inteiros com as notas dos alunos.
int *LerVetor (int *N);
 - b)** função recursiva para calcular a soma das notas obtidas pelos alunos (usar o vector).
int SomaNotas (int V[], int N);
 - c)** função recursiva para calcular a soma das notas dos alunos aprovados (usar o vector).
int SomaNotasAprovados (int V[], int N);
- 19.** As notas dos alunos da disciplina de Programação estão guardadas no ficheiro de texto "dados4.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 vector de inteiros com as notas finais (TP+T) dos alunos.

int *LerVetor (int *N);

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

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