

Programação 23/24 S2

Laboratório 2

CAP Diogo Silva

Regras

- Esta laboratório vale 1 valor (5%) da nota final da UC.
- O laboratório tem a **duração de 2 tempos letivos** e é **individual**.
- Para resolver os seguintes exercícios, podem ser usados:
 - computador
 - ambiente de desenvolvimento, sem assistências configuradas (e.g. Github Copilot)
 - * **o ambiente de desenvolvimento é local** (e.g. CodeBlocks), **sem recurso à internet** (e.g. Replit)
 - folha de rascunho
 - caneta ou lápis
 - folha de consulta “cheatsheet” (impressa ou em PDF)
- Em cima da mesa de trabalho não devem estar quaisquer outros objetos, à excepção de uma garrafa de água, se desejado.
- **Não podem ser consultados ou usados quaisquer outros recursos.**
- Lê o enunciado com atenção e usa o código fornecido (quando aplicável).
-
-
- **CONDIÇÃO MINÍMA PARA ACEITAÇÃO: O CÓDIGO COMPILA E CORRE**
-
-
- Se a solução submetida não compilar com sucesso, a nota final é 0.
- **O incumprimento das regras supracitadas levará à anulação do laboratório e consequente atribuição de pontuação 0.**
- **Escreve o NIP e NOME (primeiro e último) em forma de comentário no final do ficheiro**, e.g. “134569 Isaac Newton”.
- Submete a solução no Moodle.

Problema

- Contexto do problema
 - Para uma determinada turma de alunos, pretende-se publicar as notas relativas a uma avaliação.
 - Pretende-se que
 1. os alunos possam consultar os seus resultados
 2. os alunos possam visualizar a distribuição das notas
 3. os alunos mantenham a sua privacidade
 - Assim, deliberou-se que cada aluno iria receber um identificador único.
 - O identificador é constituído por 2 palavras, e.g. “boring einstein”, provenientes de 2 listas de palavras distintas (uma palavra de cada lista).
 - Cada lista de palavras está guardada num ficheiro de texto próprio, nomeadamente `adjectives.txt` e `names.txt`.

- A primeira linha de cada ficheiro de texto contém o número de palavras (e por conseguinte linhas) que o ficheiro contém, seguido das palavras, uma por linha. Assume-se que cada palavra tem no máximo 100 caracteres.
- Os 2 exercícios deste laboratório implementam uma programa que resolve este problema.
- A função `main` terá uma componente dos dois exercícios. Usar só uma `main` - como sabem o C não permite mais que uma `main`.
- Não é necessário pedir quaisquer dados ao utilizador.

Exercício 1 - Ler lista de ficheiros [50%]

1. Cria uma função que:

- Recebe
 - Um apontador de ficheiro (já aberto)
 - Um vetor de strings e um inteiro
 - Correspondente ao tamanho do vetor.
- Percorre o ficheiro e guarda as palavras no vetor de strings.
- Devolve o número de palavras lidas.
- Ignora a primeira linha do ficheiro, que contém o número de palavras.
- Pára de ler palavras quando atinge o fim do ficheiro ou quando atinge o tamanho máximo do vetor.

2. Na `main`, o programa:

- Abre os ficheiros `adjetives.txt` e `names.txt` (atenção ao modo de abertura);
- Lê o número de palavras de cada ficheiro (inteiro na primeira linha);
- Cria
 - 1 vetor de strings com o tamanho adequado para a lista do ficheiro `adjetives.txt`;
 - 1 vetor de strings com o tamanho adequado para a lista do ficheiro `names.txt`;
- Chama a função do ponto 2 para cada ficheiro e o correspondente vetor de strings;
- Fecha os ficheiros depois de os ler.

Exercício 2 - Criar e gravar identificadores [50%]

1. Escreve uma função que:

1. Recebe

- Um vetor de strings com os adjetivos
- Um vetor de strings com os nomes
- O número de adjetivos
- O número de nomes
- O número de alunos (i.e. n.º de identificadores a criar)
- O nome do ficheiro onde se pretendem guardar os identificadores

2. Abre o ficheiro num modo apropriado para escrita de texto;

3. Cria e guarda um identificador único para cada aluno, constituído por 2 palavras, uma de cada vetor, e.g. “boring einstein”

- Estratégia para criar identificador único:
 - Percorrer vetor de nomes com um salto entre nomes igual ao último algarismo do teu NIP, e.g.
 - * Se a lista de nomes for {"wozniak", "vaughan", "tharp", "swartz", "stonebraker", "varahamihira", "wilbur", "torvalds", "noether", "mclaren"}, e o NIP for 136787;
 - * Como saltamos de 7 em 7 (último algarismo do NIP) até ao final da lista, e começamos no 0, só apanhamos os nomes com índice 0 e 7, i.e. “wozniak” e “torvalds”;
- Usar a mesma estratégia para escolher os adjetivos, mas com um salto igual ao penúltimo algarismo do teu NIP (no exemplo acima seria um salto de 3);
- Os algarismos do NIP são escritos diretamente no código - não pedir dados ao utilizador;

- Concatena a palavra escolhida do vetor de adjetivos com a palavra escolhida do vetor de nomes para criar o identificador, e escreve-o no ficheiro de destino.
 - Ajuda - a concatenação pode ser feita de várias formas
 - * usar `strcat` para concatenar progressivamente cada palavra (nome ou adjetivo), e um espaço entre elas, a uma string final;
 - * escrever progressivamente cada palavra no ficheiro de destino;
 - * entre outras...

2. Na `main`, o programa:

- Chama a função do ponto 1 com:
 - Os vetores de adjetivos e nomes lidos no exercício 1;
 - O nome de ficheiro deve ser “`identificadores_NIP.txt`”, em que NIP deve ser substituído pelo teu NIP, e.g. “`identificadores_136787.txt`”
 - O número de identificadores a criar é 50;
- Se o exercício 1 não foi resolvido, os vetores `nomes_ajuda` e `adjetivos_ajuda` definidos no final do código dado podem ser usados.