

Trabalho Prático nº 8: Strings

Observações:

- O trabalho deve ser desenvolvido no editor VSCode, com integração ao CodeInsights.
- É importante que identifique corretamente o exercício que está a resolver (exemplo: @aid 8.1) e que utilize as palavras assinaladas a verde como nomes para as variáveis. É igualmente importante que entre `//begin_inputs` e `//end_inputs` sejam apenas declaradas e inicializadas as variáveis que correspondam a inputs do programa.
- Todas as restantes variáveis, constantes e a lógica do programa deverão estar fora deste bloco.
- Os protótipos de função, se indicados em um exercício, devem ser inseridos logo após o cabeçalho de cada programa (informação de cikey, sid e aid) e as constantes simbólicas (`#define`), se existirem.
- Todos as saídas (“outputs”) dos exercícios devem ser produzidos na função “main”, ou seja, não utilizar “printfs” dentro das demais funções, a menos que seja indicado.
- **Os trabalhos são individuais.** A cópia de programas resultará na reprovação à unidade curricular de todos os alunos envolvidos e outras penalizações previstas no Estatuto Disciplinar dos Estudantes do IPC. Também não é permitido copiar programas da internet, ou qualquer outra fonte, para exercícios iguais ou similares.
- Utilize apenas a biblioteca “stdio” e as bibliotecas e funções indicadas a cada exercício.

@aid 8.1 Escreva um programa que leia dois strings introduzidos pelo utilizador e troque o conteúdo de um pelo do outro, apresentando-os a seguir. Utilize a biblioteca de manipulação de strings do C. Recorra às variáveis **str1** e **str2**.

Exemplo de *output* para: **str1=“gato”** e **str2=“crocodilo”**:

```
primeiro string: gato - segundo string: crocodilo
apos a troca:
primeiro string: crocodilo - segundo string: gato
```

@aid 8.2 Escreva um programa que leia um string e troque todas as letras para maiúsculas. Os demais caracteres não devem ser alterados. Recorra à variável **str** e não utilize a biblioteca de manipulação de strings do C.

Exemplo de *output* para: **str=“fP-p1-2023/24”**

```
antes : fP-p1-2023/24
depois: FP-P1-2023/24
```

@aid 8.3 Escreva uma função, com o protótipo descrito abaixo, que calcule o tamanho um string, de forma similar à função `strlen`. Escreva um programa que leia um string e apresente o tamanho deste utilizando tanto o `strlen` como a sua função. Recorra à variável **str**.

```
int meu_strlen(char string[]);
```

Exemplo de *output* para: **str=“informatica”**

```
string: informatica
tamanho(strlen): 11
tamanho(minha funcao): 11
```

@aid 8.4 Escreva um programa que leia um string (com espaços) e apresente o número de letras, de números e de outros caracteres no string. Recorra à variável **str**. Utilize a função `fgets` para a leitura do string. Somente é permitido utilizar, da biblioteca de manipulação de strings do C, a função `strlen`.

Exemplo de *output* para: **str="Oliveira do Hospital 3400-124"**:

```
string: Oliveira do Hospital 3400-124  
letras: 18 - numeros: 7 - outros: 4
```

@aid 8.5 Escreva uma função, com o protótipo apresentado abaixo, que converta um número binário, expresso como um string, no seu valor decimal. O string deve ter apenas os caracteres '0' e '1'. Caso o string tenha outros caracteres, a função deve retornar o valor -1. Escreva um programa que leia um string, invoque a função descrita e apresente o valor decimal ou a mensagem de erro "entrada invalida". Recorra à variável **str**. Utilize apenas a função `strlen` da biblioteca de manipulação de string do C.

```
int binarioParaDecimal(char binario[]);
```

Exemplo de *output* para: **str="1001"**

```
binario: 1001 - decimal: 9
```

@aid 8.6 Escreva uma função, com o protótipo apresentado abaixo, que remova caracteres de um string, a partir de uma posição definida. Escreva um programa que leia um string, com a função `fgets`, (variável **str**), a posição (variável inteira **pos**) e o número de caracteres (variável inteira **num**), invoque a função descrita, e apresente o conteúdo do string após esta retornar. Somente é permitido utilizar, da biblioteca de manipulação de strings do C, a função `strlen`.

```
void removeString(char string[], int posicao, int num_chars);
```

Exemplo de *output* para: **str="programa em C", pos=9 e num=3**

```
antes : programa em C  
depois: programa C
```