

Tarefa – Arquivos binários

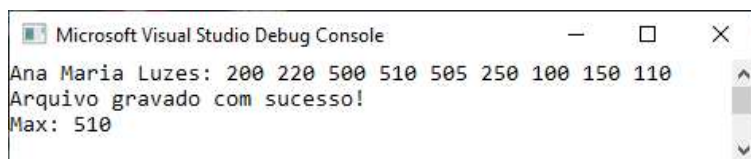
Suponha que a coleta de dados por um drone tem o seguinte registro em um arquivo texto **dadosDrone2.txt** com uma única linha contendo *nome_do_operador: lista_de_valores_inteiros* (atenção nos dois pontos logo após o nome). Por exemplo, para 9 valores:

```
Ana Maria Luzes: 200 220 500 510 505 250 100 150 110
```

Considere que o tamanho da lista (9, no exemplo acima) é fixo e conhecido. Nesta tarefa, você vai ler o arquivo e gravar os valores da lista em um arquivo binário cujo nome é **stream_nomeDoOperador.dat**. O nome do operador é o nome lido do arquivo **dadosDrone2.txt** com `_` no lugar dos brancos. No exemplo acima, o arquivo gravado é **stream_Ana_Maria_Luzes.dat** e contém 9 inteiros. Escreva uma função auxiliar para trocar brancos por `_`.

Após gravar o arquivo `.dat`, você deve fechá-lo e depois abri-lo para leitura. Uma vez aberto, leia-o e encontre o maior valor (usando uma função auxiliar).

Exemplo:



```
Microsoft Visual Studio Debug Console
Ana Maria Luzes: 200 220 500 510 505 250 100 150 110
Arquivo gravado com sucesso!
Max: 510
```

Para ler e escrever arquivos, use a mesma função auxiliar `openFile` usada na tarefa anterior:

```
FILE * openFile(const char * file, const char * mode)
```

que abre o arquivo (`file`) no modo solicitado (leitura ou escrita), usando a **fopen**, e testa se a operação foi bem sucedida (se não for, imprima uma mensagem e aborte com **exit(1)**).

Use obrigatoriamente `while (!feof(fin))` para ler o arquivo¹, mesmo sendo um arquivo de uma única linha.

Dica1: para escrever e ler binário, use os modos `"wb"` e `"rb"` respectivamente.

Dica2: para escrever binário, use a função `fwrite(ptr, size, n, fp)`, onde `ptr` aponta para o bloco de memória que contém os itens a serem escritos, `size` especifica o número de bytes de cada item a ser escrito, `n` é o número de itens a serem escritos e `fp` é o ponteiro para o arquivo. Por exemplo, se `int a[] = { 10, 20, 30};`, então `fwrite(a, sizeof(int), 3, fout)` escreve os 3 elementos de `a` no arquivo representado por `fout`.

Dica3: para ler um arquivo binário, use a função `fread(ptr, size, n, fp)`, que funciona semelhante ao caso do `fwrite`. Por exemplo, se `int a[5];`, então `fread(a, sizeof(int), 5, fin)` lê 5 valores inteiros do arquivo `fin` e armazena-os no vetor `a`.

Dica4: não é o caso, nesta tarefa, de você gravar uma simples variável em um arquivo binário. Mas, para você entender a função `fwrite` melhor, um exemplo de gravação é: se `int x = 500;`, então `fwrite(&x, sizeof(int), 1, fout)` escreve o valor da variável `x` no arquivo.

¹ Somente quando uma operação de leitura encontra um "end of file" é que o **feof()** retorna verdade (i.e. retorna $\neq 0$). Então, o `while` ainda entra uma última vez para que **fscanf** encontre "end of file" e retorne **EOF** (usualmente -1).