

Guião aula07 - Gil Guedes 125031

1 - a - O writePeople vai escrever o nome, idade e altura no ficheiro que for dado no argumento. O readPeople vai ler o ficheiro que o writePeople criou, e imprimi-lo no terminal

b -

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

typedef struct
{
    int age;
    double height;
    char name[64];
} Person;

void printPersonInfo(Person *p)
{
    printf("Person: %s, %d, %f\n", p->name, p->age, p->height);
}

int main (int argc, char *argv[])
{
    FILE *fp = NULL;
    int i, numPeople;
    Person p;

    /* Validate number of arguments */
    if(argc != 2)
    {
        printf("USAGE: %s fileName\n", argv[0]);
        return EXIT_FAILURE;
    }

    /* Ask user for number of people */
    printf("Quantas pessoas pretende armazenar? ");
    if(scanf("%d", &numPeople) != 1 || numPeople <= 0)
    {
        printf("Erro: Número de pessoas inválido!\n");
        return EXIT_FAILURE;
    }

    /* Open the file provided as argument */
    errno = 0;
    fp = fopen(argv[1], "wb");
    if(fp == NULL)
    {
```

```

        perror ("Error opening file!");
        return EXIT_FAILURE;
    }

    /* Get information for each person and write to file */
    for(i = 0; i < numPeople; i++)
    {
        printf("\n--- Pessoa %d ---\n", i + 1);

        printf("Nome: ");
        if(scanf("%63s", p.name) != 1)
        {
            printf("Erro ao ler o nome!\n");
            fclose(fp);
            return EXIT_FAILURE;
        }

        printf("Idade: ");
        if(scanf("%d", &p.age) != 1)
        {
            printf("Erro ao ler a idade!\n");
            fclose(fp);
            return EXIT_FAILURE;
        }

        printf("Altura (em metros): ");
        if(scanf("%lf", &p.height) != 1)
        {
            printf("Erro ao ler a altura!\n");
            fclose(fp);
            return EXIT_FAILURE;
        }

        /* Write person data to file */
        if(fwrite(&p, sizeof(Person), 1, fp) != 1)
        {
            printf("Erro ao escrever no ficheiro!\n");
            fclose(fp);
            return EXIT_FAILURE;
        }

        printf("Pessoa adicionada: %s, %d anos, %.2f m\n", p.name, p.age,
p.height);
    }

    fclose(fp);
    printf("\n%d pessoas foram armazenadas com sucesso no ficheiro '%s'.\n",
numPeople, argv[1]);

    return EXIT_SUCCESS;
}

```

2 - a - O ficheiro 'testOpt.c' mostra como o 'getopt()' processa opções curtas, com e sem argumentos, gerindo flags, valores e erros para entradas inválidas ou incompletas.

3 - a - O ficheiro 'testLongOpt.c' demonstra que o 'getopt_long()' processa corretamente opções curtas e longas, com ou sem argumentos, mapeando-as adequadamente e validando entradas especiais.