

TURMA: A

Nome: _____

No. USP: _____

- 1) [6.0] Considere o jogo papel(*paper*)-pedra(*rock*)-tesoura(*scissors*) e o arquivo `p_r_s.h` apresentado em aula e reproduzido abaixo, em que os protótipos das funções bem como seus próprios nomes explicam suas funcionalidades.

```
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
typedef enum {paper,rock,scissors,
game,help,instructions,quit} p_r_s;
typedef enum {win, lose, tie, error} outcome;
outcome compare(p_r_s player_choice,
p_r_s machine_choice);
void wrt_final_status(int win_cnt, int lose_cnt);
void wrt_game_status(int win_cnt, int lose_cnt, int tie_cnt);
void wrt_help(void);
void wrt_instructions(void);
void report_and_tabulate(outcome result,
int * win_cnt_ptr, int *lose_cnt_ptr, int *tie_cnt_ptr);
p_r_s selection_by_machine(void);
p_r_s selection_by_player(void);
```

- a. Elabore o código C das funções `selection_by_machine()` e `compare()`;
- b. Faça um pseudocódigo para o programa principal, baseando-se nas funções disponíveis em `p_r_s.h`.
- 2) [2.0] Faça um programa em C (cujo executável possui nome `ordena2c`) que receba n palavras pela linha de comando e as ordene lexicograficamente pela segunda letra de cada palavra.
Exemplo para $n=3$:
`./ordena2c roupa bala legal`
Saída no terminal padrão:
`bala legal roupa`
Utilize a função `qsort()` disponível na `stdlib.h`, cujo protótipo é:
`void qsort(void *array, size_t n_els, size_t el_size, int cmp(const void *, const void *));`
A função `cmp()` deve retornar `-1`, `0`, `1`.
- 3) [2.0] Faça uma função em C que receba imagens de um filme e as salve em um mesmo arquivo. Cada imagem recebida é uma matriz de 1800×1200 "pixels", que possuem os seguintes parâmetros: cor (entre 1.048.576 diferentes cores), brilho (64 níveis) e contraste (64 níveis). As 5 melhores implementações desta questão em termos de economia de memória receberão:
- +2.0 pontos;
 - +1.0 ponto;
 - +0.5 ponto;
 - +0.3 ponto;
 - +0.2 ponto.

BOA SORTE!