



```
return 0; }
```

## Prova 1 - Tipo 1

2024.2

- 1) (2 pts) Considere a seguinte função em C para calcular o fatorial de um número inteiro. Essa função pode retornar uma saída inesperada? Se sim, qual seria o problema e como ele pode ser corrigido?

```
int fat(int n) {  
    int fatorial = 1;  
    for (int i = 1; i <= n; i++){  
        fatorial *= i; }  
    return fatorial; }
```

- 2) (2 pts) O código a seguir solicita ao usuário um número inteiro positivo  $n$  e calcula a soma dos números ímpares de 1 até  $n$ . O código pode ser otimizado para melhorar sua eficiência<sup>1</sup>? Se sim, qual seria a modificação apropriada?

```
int main() {  
    int n, soma = 0, i = 1;  
    scanf("%d", &n);  
    while (i <= n) {  
        if (i % 2 != 0)  
            soma += i;  
        i++; }  
    printf("%d", soma);  
}
```

<sup>1</sup> Executar o código em um número menor de passos.

- 3) (6 pts) Crie uma função que retorne 1 se o número inteiro fornecido for **simétrico** em relação ao número de seus dígitos (isto é, se o número lido da esquerda para a direita for igual ao número lido da direita para a esquerda). Caso contrário, a função deve retornar 0.

Implemente a função

```
int is_symmetric(int x)
```

**Exemplo 1:**

Entrada:  $x = 12321$

Saída: 1

*Explicação: O número 12321 é lido da mesma forma da esquerda para a direita e da direita para a esquerda.*