

## Prática com a Linguagem C

### Informações

- Trabalho **em duplas**;
- Entregar em `<http://trab.dc.unifil.br/moodle/>`
- Peso deste trabalho no bimestre: **2** em 10.

## 1 Instruções

O objetivo deste trabalho é que todos se habituem à Linguagem C e às ferramentas envolvidas em sua utilização. As atividades são todas de programação, porém não é cobrado nada além do que já viram em disciplinas do primeiro ano. A dificuldade estará concentrada justamente no aprendizado dos recursos da Linguagem C, e requererá pesquisas, de preferência em livros<sup>1</sup>, mas também na Internet. Portanto, a pesquisa sobre funções, recursos e utilização de C faz parte do trabalho, e **não é esperado que nenhum aluno seja capaz de realizar este trabalho sem recorrer ativamente a pesquisa.**

### 1.1 Ferramentas

Todas as atividades desta lista deverão ser feitas com as seguintes ferramentas:

- Programados na *Linguagem C*;
- Editados com *vim*, *emacs* ou *Gedit*;
- Compilados com *gcc* ou *clang*;
- Projeto gerenciado com *make* e *MAKEFILE*;

Para entrega do trabalho, empacotar e compactar todos os arquivos fontes e o *MAKEFILE*, excluindo binários e outros desnecessários para compilação. Só serão aceitos trabalhos entregues no formato *.tar.gz*, utilizando a ferramenta *tar* do terminal *bash*.

Além disso, o trabalho deverá ser compilado corretamente com uma simples chamada ao *make* no terminal *bash*, no diretório do projeto. Essa condição é **fundamental** para o trabalho ser corrigido!

## 2 Atividades

1. (5 pontos) Faça um programa, denominado `atividade1`, para receber 3 valores inteiros do usuário e mostrar a sua média (que pode não ser inteira). Para tal, utilize as funções `printf` e `scanf`.

---

<sup>1</sup>Verificar o Plano de Ensino da disciplina para a bibliografia complementar.

2. (10 pontos) Faça um programa, denominado `atividade2`, que recebe um símbolo de operação do usuário (+, -, / ou \*) e dois números reais. O programa deve retornar o resultado da operação recebida sobre estes dois números.
3. (10 pontos) Faça um programa, denominado `atividade3`, que receba uma frase do usuário e a escreva ao contrário. Crie uma função para isso.
4. (20 pontos) Faça um programa, denominado `atividade4`, que recebe uma palavra ou frase do usuário e indique se ela é um palíndromo<sup>2</sup> ou não. A palavra ou frase deverá ser passada como argumento de linha de comando<sup>3</sup>. Implemente duas abordagens distintas para a verificação:
  - (a) Comparação caractere por caractere;
  - (b) Utilizando a função `strcmp` e a que você criou no exercício 3.
5. (10 pontos) Faça um programa, denominado `atividade5`, que sorteie<sup>4</sup> um número de 0 a 100 e que permita que o usuário (sem conhecer o número sorteado) tente acertar. Caso não acerte, o programa deve imprimir uma mensagem informando se o número sorteado é maior ou menor que a tentativa feita. Ao acertar o número, o programa deve imprimir a quantidade de tentativas feitas.
6. (30 pontos) No código-fonte `data_hora.c`, faça o que se pede:
  - (a) Crie uma `struct hora`;
  - (b) Crie uma `struct data`;
  - (c) Programe a função `calcula_duracao_jogo`, que recebe, por parâmetro, a hora de início e a hora de término de um jogo. A função deve retornar, a duração do jogo em minutos, considerando que o tempo máximo de duração de um jogo é de 24 horas e que o jogo pode começar em um dia e terminar no outro;
  - (d) Programe a função `data_valida`, que recebe como uma data e devolve 1 se a data for válida e 0 em caso contrário. Fique atento para datas em anos bissextos (Um ano é bissexto se `(ano % 4 == 0 && (ano % 100 != 0 || ano % 400 == 0))`).
  - (e) Programe a função `passar_dia`, que tem como parâmetro uma data e a modifica de forma que ela represente o dia seguinte;
  - (f) Programe a função `passar_lote_dias`, que leia um inteiro positivo `n` e uma sequência de `n` datas e retorne, para cada data, o dia seguinte.

Utilize estas funções no seu código `main.c` para testá-las e escreva na tela os resultados dos testes.

7. (15 pontos) No código-fonte `lista.c`, faça o que se pede:

---

<sup>2</sup>Ver <<http://pt.wikipedia.org/wiki/Pal%C3%ADndromo>>.

<sup>3</sup>Procurar por *argc* e *argv*.

<sup>4</sup>Utilize as funções *srand* e *rand*.

- (a) Crie uma `struct elemento`, que contém um valor inteiro e um apontador para `struct elemento`;
- (b) Programe uma função que receba como parâmetro um arranjo de inteiros e retorne uma lista de inteiros utilizando `struct elemento`;
- (c) Programe uma função que receba um `struct elemento` como parâmetro e retorne um arranjo de inteiros.

Utilize estas funções no seu código `main.c` para testá-las e escreva na tela os resultados dos testes.