Trabalho prático de CI1001
TP2
Departamento de Informática/UFPR

1 Sobre a entrega do trabalho

São requisitos para atribuição de notas a este trabalho:

- Uso de um arquivo makefile para facilitar a compilação. Os professores rodarão "make" e deverão obter o arquivo executável funcional com a sua solução. Este executável, cujo nome deverá ser tp2, deverá estar no subdiretório tp2;
- Ao compilar, incluir pelo menos `-Wall`. Se não compilar, o trabalho vale zero. Haverá desconto por cada *warning*;
- Arquivo de entrega:
  - Deve estar no formato tar comprimido (.tgz);
  - O tgz deve ser criado considerando-se que existe um diretório com o nome do trabalho. Por exemplo, este trabalho é o tp2;
  - Então seu tgz deve ser criado assim:
    - Estando no diretório tp2, faça:
    - `cd ..`
    - tar zcvf tp2.tgz tp2
  - Desta maneira, quando os professores abrirem o tgz (com o comando tar zxvf tp2.tgz) terão garantidamente o diretório correto da entrega para poderem fazer a correção semi-automática.
  - O que colocar no tgz? Todos os arquivos que são necessários para a compilação, por isso se você usa arquivos além dos especificados, coloque-os também. Mas minimamente deve conter todos os arquivos `.c`, `.h` e o makefile;
  - Os professores testarão seus programas em uma máquina do departamento de informática (por exemplo, cpu1), por isso, antes de entregar seu trabalho faça um teste em máquinas do dinf para garantir que tudo funcione bem.

2 Objetivos

Este trabalho tem como objetivo modificar o Tipo Abstrato de Dados (TAD) para números racionais feito no TP1 para que algumas funções recebam parâmetros por ponteiros. O motivo da alteração é para que os protótipos das funções fiquem mais elegantes, pois tendo a opção de modificar o valor de uma variável recebida como parâmetro por endereço, podemos utilizar o retorno da função para retornar um código de erro.

O programa principal também é diferente para que o novo conhecimento possa ser testado.

Observem que nesta versão do `racionais.h` também foram adicionadas estas linhas:

```
#ifndef _LIBpilha_t_H
#define _LIBpilha_t_H

#endif
```

São instruções para o pré-processador e tem o efeito de evitar que um mesmo arquivo seja incluído mais de uma vez.

3 O trabalho

Você deve adaptar a sua implementação do arquivo `racionais.c` conforme o novo arquivo `racionais.h` fornecido. Observem que algumas poucas funções (as que fazem as operações e a simplificação) são diferentes da versão anterior do TP1 pois contêm parâmetros por ponteiros.

Você deve baixar o tp2.tgz anexo a este enunciado e abrí-lo para poder fazer o trabalho, pois irá precisar de todos os arquivos ali contidos:

racionais.h: arquivo (read only) de *header* com todos os protótipos das funções para manipular números racionais;

racionais.c: um esqueleto de arquivo racionais.c;

makefile: sugestão de um makefile que você pode usar.
É sua responsabilidade fazer as adaptações necessárias neste arquivo sugerido.

tp2.c: um esqueleto de arquivo tp2.c.

O arquivo `.h` não pode ser alterado. Na correção, os professores usarão os arquivos `.h` originais.

- Use boas práticas de programação, como indentação, bons nomes para variáveis, comentários no código, bibliotecas, defines... Um trabalho que não tenha sido implementado com boas práticas vale zero.
- Quaisquer dúvidas com relação a este enunciado devem ser solucionadas via email para `prog1prof@inf.ufpr.br` pois assim todos os professores receberão os questionamentos. Na dúvida, não tome decisões sobre a especificação, pergunte!
- Dúvidas podem e devem ser resolvidas durante as aulas.

4 Seu programa

No arquivo `racionais.h` foi definida uma interface para o tipo abstrato de dados *racional* que usa a mesma *struct* para os números racionais usada no TP1. Você deve implementar o arquivo `racionais.c` conforme especificado no `racionais.h` fornecido. A sua função `main` deve incluir o *header* `racionais.h` e deve implementar corretamente em C o seguinte pseudocódigo:

```
inicialize a semente randomica, uma unica vez em todo o codigo
    - sugestao: use "srand (0)" para facilitar os testes
leia um n tal que 0 < n < 100
crie um vetor de n posicoes contendo numeros racionais aleatorios
imprima este vetor
elimine deste vetor os elementos invalidos
imprima o vetor resultante
ordene este vetor
imprima o vetor ordenado
calcule e imprima a soma de todos os elementos do vetor

retorne 0
```

Imprima os elementos do vetor em uma única linha usando um único espaço em branco para separar os elementos. Ao final do vetor mude de linha.

5 Exemplo de entrada e saída

Considerando que o usuário digitou `n = 13` a saída poderia ser assim, dependendo do gerador de números aleatórios.

```
13
4/3 9 5/9 1 2/5 INVALIDO 5/4 10 9/2 0 1/10 4/5 2/3
4/3 9 5/9 1 2/5 2/3 5/4 10 9/2 0 1/10 4/5
0 1/10 2/5 5/9 2/3 4/5 1 5/4 4/3 9/2 9 10
a soma de todos os elementos eh: 5329/180
```

6 O que entregar

Entregue um único arquivo `tp2.tgz` que contenha por sua vez os seguintes arquivos:

- `racionais.h`: o mesmo arquivo fornecido, não o modifique;
- `racionais.c`: sua implementação do `racionais.h`;
- `tp2.c`: contém a função `main` que usa os racionais;
- `makefile`

Atenção: Não modifique em nenhuma hipótese o arquivo `racionais.h`. Na correção, os professores usarão o arquivo originalmente fornecido.

Bom trabalho!