MC202GH - ESTRUTURAS DE DADOS

2º Semestre de 2017

Professor: Rafael C. S. Schouery

Monitores: Yulle Glebbyo Felipe Borges (PED)

Vitor Andrietta Razoli Silva (PAD)

Lab. 01 - Representação Abstrata de Números Racionais

Peso 1

1. O Problema

Computadores convencionais possuem métodos eficientes para armazenar diferentes tipos de dados. Por exemplo, a representação computacional mais comum para os números racionais é dada em números de ponto flutuante como float e double. Entretanto, além de serem menos intuitivos, estes tipos possuem precisão limitada. Podemos representar o resultado de uma divisão do inteiro 1 pelo inteiro 3 de maneira exata como 1/3, mas este valor em ponto flutuante não existe e pode ser apenas aproximado por 0.33333… Isso pode causar problemas de precisão quando lidamos com números muito pequenos que devem ser aproximados a cada operação efetuada. Além disso, quando aprendemos aritmética sobre os números racionais, tratamos-os naturalmente com a notação de numerador e denominador, diferente do que vemos ao utilizar uma calculadora ou computador para efetuar as mesmas operações.

Com isso em mente, você foi designado para ser o monitor de um projeto inovador da Unicamp para ensinar conceitos básicos de computação integrada às aulas do ensino fundamental para alunos da rede pública de Campinas. Sua primeira tarefa é criar um tipo abstrato de dados que será utilizado pelos alunos para representar os números racionais e algumas operações básicas entre eles.

Neste TAD, um número racional será representado como um inteiro *numerador* e outro inteiro *denominador*. Você deverá implementar funções de *adição*, *subtração*, *multiplicação* e *divisão*, que recebem como parâmetros dois números racionais e retornam um racional resultante da operação invocada. Os resultados de todas as operações entre dois racionais deve ser retornado como um racional simplificado, ou seja deve-se

simplificar-lo até que o máximo divisor comum entre o numerador e denominador seja 1. Por exemplo, 4/6 deve ser simplificado para 2/3. Além destas operações, você deverá fornecer uma função de comparação que recebe números racionais A e B quaisquer (não necessariamente simplificados), e retorna 1 caso A seja maior que B, 0 caso eles sejam iguais, e -1 caso B seja maior que A. Por fim, é necessário implementar funções de entrada e saída para o TAD, logo teremos uma função que não recebe parâmetros e retorna um racional lido a partir da entrada padrão, e uma função que recebe um número racional e o imprime na forma "numerador/denominador" na saída padrão.

Para este trabalho, iremos considerar o denominador nunca pode ser 0, e o número 0 é representado por 0/x para qualquer inteiro x, ou de maneira simplificada 0/1. Caso o número racional a ser representado seja negativo, utilizaremos como padrão o sinal no numerador, logo dada uma entrada 1/-2, a impressão deve ser feita na forma -1/2.

2. Entrada

O objetivo deste laboratório é implementar um tipo abstrato de dados que será utilizado por alunos de ensino fundamental para aprender operações entre números racionais e noções sobre computação. Por isso, será necessário entregar apenas a interface e implementação do TAD, e o cliente será fornecido no SuSy. Usaremos a seguinte função cliente para testar as estruturas:

```
#include "racionais.h"
#include <stdio.h>

int main (int argc, char* argv[]) {
	int escolha, res_cmp;
	racional r1, r2, res;
	do {
		scanf("%d",&escolha);
		switch (escolha) {
			case 1: {
				r1 = racional_le();
				r2 = racional_le();
				res = racional_adiciona(r1,r2);
				racional_imprime(res);
				break;
			}
			case 2: {
```

```
                        r1 = racional_le();
                        r2 = racional_le();
                        res = racional_subtrai(r1,r2);
                        racional_imprime(res);
                        break;
                }
                case 3: {
                        r1 = racional_le();
                        r2 = racional_le();
                        res = racional_multiplica(r1,r2);
                        racional_imprime(res);
                        break;
                }
                case 4: {
                        r1 = racional_le();
                        r2 = racional_le();
                        res = racional_divide(r1,r2);
                        racional_imprime(res);
                        break;
                }
                case 5: {
                        r1 = racional_le();
                        res = racional_simplifica(r1);
                        racional_imprime(res);
                        break;
                }
                case 6: {
                        r1 = racional_le();
                        r2 = racional_le();
                        res_cmp = racional_compara(r1,r2);
                        printf("%d\n", res_cmp);
                        break;
                }
        }
    } while (escolha != 0);
    printf("Sistema encerrado");
    return 0;
}
```

As entradas serão compostas por um identificador de operação em uma linha, seguido dos valores necessários para execução da operação escolhida na linha seguinte. A tabela abaixo descreve todas as funções a requeridas:

Identificador	Nome	Descrição
1	Adição	São fornecidos dois valores inteiros que irão compor um racional **a**, e dois outros valores inteiros que irão compor um racional **b**. Imprimir um racional simplificado **c**, resultado de a + b.
2	Subtração	São fornecidos dois valores inteiros que irão compor um racional **a**, e dois outros valores inteiros que irão compor um racional **b**. Imprimir um racional simplificado **c**, resultado de a - b.
3	Multiplicação	São fornecidos dois valores inteiros que irão compor um racional **a**, e dois outros valores inteiros que irão compor um racional **b**. Imprimir um racional simplificado **c**, resultado de a * b.
4	Divisão	São fornecidos dois valores inteiros que irão compor um racional **a**, e dois outros valores inteiros que irão compor um racional **b**. Imprimir um racional simplificado **c**, resultado de a / b. Suponha b diferente de 0.
5	Simplificação	Dados dois valores inteiros que compõem o racional **a**, imprime a versão simplificada de a.
6	Comparação	São fornecidos dois valores inteiros que irão compor um racional **a**, e dois outros valores inteiros que irão compor um racional **b**. Imprimir **0** caso a seja igual a b, 1 caso a seja maior que b e -1 caso b seja maior que a.
0	Finalização	Saia do sistema.

Exemplo de entrada:

```
1
1 2 2 1
2
1 2 1 4
3
2 4 1 2
5
15 20
```

```
6
1 3 2 6
0
```

3. Saída

Para cada operação na entrada, é necessário uma resposta correspondente em uma linha diferente da saída. Para as operações de 1 até 5, é esperado um racional na forma "numerador/denominador", onde numerador e denominador são números inteiros. Para a operação 6, deve-se imprimir 0, 1 ou -1 de acordo com a definição da operação. Por fim, para a operação 0, deve-se imprimir "Sistema encerrado.".

Exemplo de saída:

```
5/2
1/4
1/4
3/4
0
Sistema encerrado.
```

4. Informações

- Este laboratório possui peso 1.
- Não há um número máximo de submissões.
- No início de cada arquivo a ser submetido, insira um comentário com seu nome, RA e uma breve descrição do conteúdo do arquivo.
- Apenas comentários no formato `/* comentário */` serão aceitos. Comentários com `//` serão acusados como erros pelo SuSy.
- Além da corretude do código submetido, iremos avaliar também sua indentação. Escolha entre espaços ou tabs para identar seu código e seja

coerente em todos os arquivos. Indentações utilizando espaços e tabs intercalados, ou trechos não indentados corretamente irão resultar em uma penalização de um ponto na nota.

- A submissão da sua solução deverá conter múltiplos arquivos:
    - **racionais.h**: interface da estrutura de dados
    - **racionais.c**: implementação da interface
- O cliente, ou seja, a implementação da função main será dado, portanto não deverá ser submetido nenhum arquivo com esta função ao SuSy.
- Será disponibilizado um **Makefile** para este trabalho na página do laboratório:
    - Para compilar seu projeto, basta navegar até o diretório do projeto e utilizar o comando ‘make’ no terminal do Linux.
    - Veja mais instruções na página da disciplina.