

Prova 1

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

02 de maio de 2017

Nome:

_____ escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca `prova1.h`), função `main`, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.
- As funções implementadas no módulo `prova1.h` podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo `prova1.h`, considere que ela está implementada de forma correta.

1. (4 points)

Escreva uma função de nome `ints2Exceed` que recebe dois inteiros A e B como parâmetros e retorna o **menor** número de inteiros que devem ser somados em sequência a partir de (e incluindo) A de modo que a soma exceda B . Exemplo: se $A = 3$ e $B = 14$, então a menor quantidade de números em sequência a partir de 3 que quando somados supera 14 é 3, 4, 5, 6, o que faz a função retornar, nesse caso, 4 (a quantidade de números dessa sequência).

2. (3 points) Escreva um **programa** que lê dois números inteiros do teclado X e Z e imprime na tela o **menor** número de inteiros que devem ser somados em sequência a partir de (e incluindo) X de modo que a soma exceda Z . Para isso, use a função `ints2Exceed` do exercício anterior. O seu programa deve garantir que Z seja maior que X , ou seja, o usuário deve pedir valores de Z para o usuário até que essa condição seja satisfeita.

3. (3 points) Escreva uma função que recebe um número ponto flutuante x e o arredonda a partir da seguinte regra:

- se a parte fracionária de x for menor que 0.25, mantenha a sua parte inteira como está e arredonde a sua parte fracionária para 0.0;
- se a parte fracionária de x for maior ou igual a 0.25 e menor que 0.75, mantenha a sua parte inteira como está e arredonde a sua parte fracionária para 0.5;
- se a parte fracionária de x for maior ou igual a 0.75, incremente a sua parte inteira de 1 unidade e arredonde a sua parte fracionária para 0.0;

4. (4 points) Escreva um procedimento (função que não retorna nada) de nome `arredondaNotas` que recebe três endereços de memória que armazenam pontos flutuantes: `endNota1`, `endNota2` e `endNota3`. O seu procedimento deve acessar esses endereços e arredondar os pontos flutuantes armazenados neles usando a função do exercício anterior. Exemplo: se `endNota1=FF2201`, `endNota2=FF2204` e `endNota3=FF2208`, e nesses endereços estiverem armazenados os valores 13.12, 19.65 e 23.80, respectivamente, então você deve, ao final do seu procedimento, armazenar nesses endereços os valores 13.0, 19.5 e 24.0, respectivamente.

5. (2 points) Complete o programa abaixo, que deve ler três notas de alunos do teclado e imprimir os seus valores arredondados.

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

```
#include _____
```

```
void main() {  
    float nota1, nota2, nota3;  
    scanf("%f %f %f", &nota1, &nota2, &nota3);  
  
    arredondaNotas(_____);  
    printf("\n%f %f %f", nota1, nota2, nota3);  
}
```