

“Prova Resgate” de Programação de Computadores

Professor: Carlos Camarão

9 de Novembro de 2004

1. (5 pontos) Defina uma função *somat* que receba como parâmetro um número inteiro positivo n e retorne um valor aproximado do somatório:

$$\frac{1}{1} + \left(-\frac{2}{4}\right) + \frac{3}{9} + \left(-\frac{4}{16}\right) + \frac{5}{25} + \dots$$

usando n parcelas.

Escreva um programa para teste dessa função. O programa deve ler um valor inteiro n , terminar a execução se n for menor ou igual a zero e, caso contrário, chamar a função *somat* passando n como argumento, imprimir o valor retornado e repetir o processo (de leitura seguida da chamada a *somat* e impressão do valor retornado).

Dica: Não se esqueça de evitar a realização de divisão entre dois números inteiros (de tipo `int`). Lembre-se que, para `num` e `denom` do tipo `int` e, por exemplo, `num = 2` e `denom=4`, o valor de `num/denom` é igual a zero.

2. (5 pontos) Escreva definições para as funções *soma* e *somaR*, tais que, ao receber como argumentos dois números inteiros a e b , retornem o valor $a + b$, mas suponha que as operações de adicionar e de subtrair mais de uma unidade não são disponíveis (i.e. ou seja, você deve basear a sua definição no uso das operações mais simples de incrementar 1 e decrementar 1). *soma* deve ser baseada no uso de um comando de repetição, e *somaR* deve ser baseada em chamadas recursivas.

Escreva um programa para teste dessas funções. O programa deve ler dois valores inteiros, terminar a execução se ambos forem iguais a zero e, caso contrário, chamar e imprimir o resultado de *soma* e *somaR* para os valores lidos, e repetir o processo (de leitura seguida do cálculo e impressão da soma dos valores lidos).

3. (5 pontos) Defina uma função *decBin* que receba um número inteiro não-negativo como argumento e retorne uma cadeia de caracteres que é igual à representação desse número em notação binária.

Por exemplo, ao receber o número inteiro 8, a função deve retornar "1000".

Defina também uma função *binDec* que receba como argumento uma cadeia de caracteres que representa um número inteiro não-negativo em notação binária e retorne o valor inteiro (de tipo `int`) correspondente, em notação decimal.

Por exemplo, ao receber a cadeia de caracteres "1000", a função deve retornar o valor 8.

Escreva um programa para teste dessas funções. O programa deve ler um valor inteiro n , terminar a execução se esse valor for menor que zero e, caso contrário, chamar e imprimir o resultado de $binDec(decBin(n))$, e repetir o processo (de leitura de n seguida de cálculo e impressão de $binDec(decBin(n))$, se $n \geq 0$).

Não use a operação de exponenciação para definição de $binDec$! Escreva $binDec$ realizando a operação de conversão da cadeia de bits da esquerda (dígitos mais significativos) para a direita (dígitos menos significativos); use para isso uma variável n que representa o resultado da conversão, e uma variável i para percorrer os índices da cadeia de bits; para cada novo dígito encontrado, defina n como a multiplicação do valor (anterior) de n por 2 somado ao valor inteiro correspondente ao bit na posição i , e incremente i ; repita até que i seja maior ou igual ao tamanho da cadeia.