ED1 2024/2: Lista de Exercícios n.º 1
Estrutura de Dados I
Professor(es): Abrantes Araújo Silva Filho Entrega: 2024-08-11 23:59:59

Exercícios de programação: revisão de C

Utilize a linguagem **C** para resolver os problemas abaixo (consulte o modo de entrega ao final do documento). Recomenda-se utilizar a máquina virtual da disciplina para a programação.

(a) Escreva um programa que lê um número inteiro positivo N e então calcule e exiba na tela a soma dos primeiros N inteiros ímpares. Por exemplo: se $N = 4$, seu programa deve exibir 16, que é $1 + 3 + 5 + 7$.

(b) Escreva um programa que lê uma lista de inteiros informados pelo usuário (um número a cada linha) até que o usuário informe 0 como um valor sentinela. Quando o sentinela for digitado, seu programa deve exibir o maior valor da lista.

(c) Escreva um programa que lê um número inteiro informado pelo usuário e imprime um número inteiro com a mesma quantidade de algarismos, mas em ordem reversa. Se o usuário informar "1234567890", seu programa deve imprimir "0987654321". Note que zeros à esquerda devem ser impressos.

(d) Os matemáticos gregos tinham um interesse especial em números que eram iguais à soma de seus divisores próprios (um divisor próprio de N é qualquer divisor menor do que o próprio N). Eles chamavam esses números de **números perfeitos**. Por exemplo, 6 é um número perfeito pois ele é a soma de 1, 2 e 3, que são os inteiros menores do que 6 que também são divisores de 6. So mesmo modo, 28 é um número perfeito pois é a soma de 1, 2, 4, 7 e 14.

Escreva o predicado `e_perfeito` que recebe um inteiro n e retorna TRUE se n for perfeito, e FALSE caso contrário. Teste sua implementação utilizando esse predicado na função `main` de seu programa, fazendo uma busca por números perfeitos entre 1 e 9999. Quando um número perfeito for encontrado, seu programa deve imprimir esse número na tela (um número por linha). As primeiras duas linhas que seu programa exibirá devem mostrar o número 6 e o número 28. Seu programa deve encontrar outros dois números perfeitos entre 1 e 9999.

(e) Cada número inteiro positivo maior do que 1 pode ser expresso como o produto de números primos. Essa fatoração é única e é chamada de **decomposição em fatores primos** (*prime factorization*). Por exemplo, o número 60 pode ser decomposto nos fatores $2 \times 2 \times 3 \times 5$, cada um dos quais é primo. Note que o mesmo número primo pode aparecer mais de uma vez em uma fatoração.

Escreva um programa que recebe um número inteiro $N > 1$ informado pelo usuário, e retorne a fatoração desse número. Seu programa deve imprimir a fatoração completa, utilizando asteriscos para indicar a multiplicação. Por exemplo, se o usuário informar 60, seu programa deve exibir:

```
2 * 2 * 3 * 5
```

Note que os asteriscos devem ser impressos somente entre os números.

(f) O matemático alemão Leibniz (1646–1716) descobriu um fato importante de que o número irracional π pode ser calculado utilizando-se a seguinte equação:

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots \tag{1}$$

A fórmula à direita do sinal de igualdade é uma **série infinita**; cada fração representa um termo nessa série. Se você iniciar com 1, subtrair um terço, somar um quinto, subtrair um sétimo e assim por diante, para cada um dos números ímpares, você obtém um número que é cada vez mais próximo do valor de $\pi/4$.

Escreva um programa que calcule uma aproximação do valor de π, utilizando os primeiros dez mil termos na série de Leibniz. No final seu programa deve imprimir o valor aproximado de π com 20 casas decimais.