ED1 2025/1: Lista de Exercícios n.º 1
Estrutura de Dados I
Professor(es): Abrantes Araújo Silva Filho Entrega: 2025-02-18 07:15

As questões e exercícios de programação a seguir foram retiradas do livro "*Programming Abstractions in C: A Second Course in Computer Science*", de Eric S. Roberts (Addison-Wesley, 1998).

1 Questões de Revisão

Responda às questões abaixo, de forma **manuscrita**, em papel almaço. Entregue diretamente ao professor na data indicada.

(a) Qual é a diferença entre um arquivo fonte e um arquivo objeto?

(b) Em um comando `#include`, o nome da *header file* da biblioteca pode estar entre aspas duplas ou entre os sinais de menor do que e maior do que. Qual a diferença entre essas formas?

(c) Qual o nome da função que deve ser definida em todo programa C?

(d) Qual a diferença entre variáveis locais estáticas e não estáticas? E entre variáveis globais estáticas e não estáticas?

(e) O que é um tipo de dado?

(f) Qual a diferença entre os tipos de dados `int` e `unsigned int`?

(g) Liste todos os possíveis valores do tipo `_Bool`.

(h) Indique os valores e os tipos das seguintes expressões:

- `2 + 3`
- `19 / 5`
- `19.0 / 5`
- `19 % 5`
- `2 % 7`

(i) Ao aplicar as regra apropriadas de procedência e associatividade, qual o resultado das seguintes expressões?

- `6 + 5 / 4 - 3`
- `2 + 2 * (2 * 2 - 2) % 2 / 2`
- `10 + 9 * ((8 + 7) % 6) + 5 * 4 % 3 * 2 + 1`
- `1 + 2 + (3 + 4) * ((5 * 6 % 7 * 8) - 9) - 10`

(j) O que significa "avaliação em curto-circuito"?

(k) Descreva em Português a operação geral da sentença de controle `switch`.

(l) O que é um valor sentinela? Para que é comumente utilizado?

(m) Que linha de controle do loop `for` você deve usar para:

- Contar de 1 até 100
- Contar de 7 em 7, iniciando em 0 até que o número tenha 3 algarismos
- Contar de 2 em 2, iniciando em 100 até 0

(n) Qual a diferença entre função, procedimento e predicado?

(o) Quais as características de uma função pura?

2 Exercícios de Programação

Utilize a linguagem **C** para resolver os problemas abaixo. Se necessário, utilize as funções das bibliotecas CRpaic e CSLIB.

(a) Escreva um programa que recebe a temperatura em graus Celsius do usuário e exibe a temperatura em graus Fahrenheit. Lembre-se de que a conversão é dada por:

$$F = \frac{9}{5} \times C + 32 \tag{1}$$

(b) Escreva um programa que converte uma distância em metros informada pelo usuário para a distância padrão dos Estados Unidos, em pés e polegadas. Os fatores de conversão que você precisa são os seguintes:

- 1 polegada = 0,0254 metros
- 1 pé = 12 polegadas

(c) Escreva um programa que recebe um número inteiro positivo N do usuário e calcula e exibe na tela a soma dos primeiros N inteiros ímpares. Por exemplo: se $N = 4$, seu programa deve exibir 16, que é $1 + 3 + 5 + 7$.

(d) 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.

(e) 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.

(f) 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 `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.

(g) 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.

(h) Quando um número de ponto flutuante é convertido para um número inteiro em C, o valor é truncado e a fração é descartada. Assim, quando 4.99999 é convertido em um inteiro, o resultado é 4. Em muitos casos seria útil ter a opção de **arredondar** o número de ponto flutuante para o inteiro mais próximo.

Para um número de ponto flutuante positivo x, a operação de arredondamento pode ser obtida adicionando-se 0.5 à x e, então, truncando o número para um inteiro.

Se a fração decimal de x for menor do que 0.5, o valor truncado será o inteiro menor do que x; se a fração é 0.5 ou mais, o valor truncado será o inteiro maior do que x. Como o truncamento sempre ocorre em direção à zero, números negativos devem ser arredondados subtraindo-se 0.5 e truncando, ao invés de adicionarmos 0.5.

Escreva a função `arredondar(x)`, que arredonda um número de ponto flutuante x para o inteiro mais próximo. Demonstre que sua função funciona com diversos casos de teste.

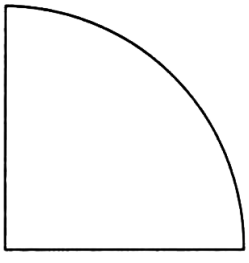
(i) O matemático alemão Leibniz (1646–1716) descobriu um fato importante de que o número irracional π pode ser calculado utilizando-se a seguinte equação:

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots \tag{2}$$

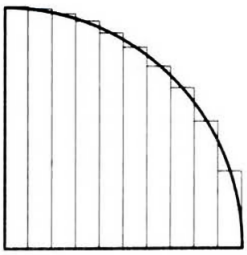
A fórmula à direita do sinal de igualdade é uma **série infinita**; cada fração representa um termo nessa série. Se você iniciar com 1, subtrair um terço, somar um quinto, subtrair um sétimo e assim por diante, para cada um dos números ímpares, você obtém um número que é cada vez mais próximo do valor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

(j) Você também pode aproximar π através da aproximação do cálculo da área limitada por um arco circular. Considere o seguinte arco que tem raio r igual a 2.0 centímetros e corresponde a um quarto de círculo:



A partir da fórmula para a área de um círculo, você pode facilmente determinar que a área do quarto de círculo deve ser π centímetros quadrados. Você também pode aproximar a área computacionalmente através da adição das áreas de uma série de retângulos, onde cada retângulo tem uma largura fixa e a altura é escolhida de forma que o círculo passe pelo ponto médio do topo do retângulo. Por exemplo: se você dividir a área em 10 retângulos da esquerda para a direita, você obtém o seguinte diagrama:



A soma das áreas dos retângulos se aproxima da área do quarto de círculo. Quanto mais retângulos você utilizar, melhor a aproximação.

Para cada retângulo, a largura w é a constante derivada pela divisão do raio pelo número de retângulos. A altura h, por outro lado, varia dependendo da posição do retângulo. Se o ponto médio do retângulo na direção horizontal é dada por x, a altura do retângulo pode ser calculada usando-se a fórmula da distância:

$$h = \sqrt{r^2 - x^2} \tag{3}$$

A área de cada retângulo é simplesmente $h \times w$.

Escreva um programa que calcule a área do quarto de círculo dividindo-o em 100 retângulos.