Aula prática 7

Comandos de repetição — while

Resumo

Nesta aula você desenvolverá algumas aplicações para treinar o uso do comando `while`.

Sumário

1 Resolvendo problemas

Tarefa 1: Sequência de Collatz

A conjectura de Collatz, também conhecida como conjectura $3n + 1$, foi proposta pelo matemático Lothar Collatz, em 1937. Para explicar essa conjectura, considere o seguinte processo que descreve como obter a sequência de Collatz para um número inteiro $n > 0$:

> Se n for par, divida n por 2, obtendo $n/2$; se n for ímpar, multiplique n por 3 e some 1, obtendo $3n + 1$. Repita esse processo para o valor obtido, e assim sucessivamente, até que o valor obtido seja 1.

A tabela seguinte mostra algumas sequências de Collatz.

n	sequência de Collatz para n
5	5, 16, 8, 4, 2, 1
11	11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
12	12, 6, 3, 10, 5, 16, 8, 4, 2, 1

A conjectura é que esse processo de cálculo sempre termina: sempre se obtém, eventualmente, o valor 1, para qualquer inteiro $n > 0$ dado inicialmente. Tal conjectura nunca foi provada, mas também nunca se encontrou um exemplo em contrário.

Escreva um programa que leia um valor inteiro $n > 0$ e imprima a sequência de Collatz para n, assim como o comprimento dessa sequência. O programa deve verificar se o valor de entrada é válido, solicitando um novo valor caso não o seja.

Exemplo de execução da aplicação

```
Sequência de Collatz
--------------------
Digite um número inteiro não negativo: 11

Sequência: 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Comprimento da sequência: 15
```

Exemplo de execução da aplicação

```
Sequência de Collatz
--------------------
Digite um número inteiro não negativo: -8
Valor inválido.
Digite um número inteiro não negativo: -2
Valor inválido.
Digite um número inteiro não negativo: 0
Valor inválido.
Digite um número inteiro não negativo: 5

Sequência: 5 16 8 4 2 1
Comprimento da sequência: 6
```

Solução:

```
clear; clc;
printf("Sequência de Collatz\n");
printf("--------------------\n");
n = input("Digite um número inteiro não negativo: ");
while int(n) <> n | n <= 0 do
    printf("Valor inválido.");
    n = input("Digite um número inteiro não negativo: ");
end
cont = 1;     // quantidade de números na sequência
printf("\nSequência: %g", n);   // exibe o primeiro
while n <> 1
    if modulo(n, 2) == 0 then   // par
        n = n / 2;
    else                        // ímpar
        n = 3 * n + 1;
    end
    printf(" %g", n);
    cont = cont + 1;
end
printf("\nComprimento da sequência: %g\n", cont);
```

Tarefa 2: Valor aproximado de π

O valor de π pode ser aproximado pela seguinte série:

$$\frac{\pi}{8} = \frac{1}{1 \times 3} + \frac{1}{5 \times 7} + \frac{1}{9 \times 11} + \cdots$$

Escreva um programa que leia um valor $n \geq 1$ e calcule um valor aproximado para π usando n termos da série acima.

O seu programa deve fazer a validação da entrada. Caso o número digitado pelo usuário seja inválido, deve-se exibir uma função apropriada e solicitar ao usuário que digite novamente.

Exemplo de execução da aplicação

```
Cálculo do valor aproximado de pi
-----------------------------------
Quantidade de iterações: -5
  Valor inválido. Tente novamente.

Quantidade de iterações: 0
  Valor inválido. Tente novamente.

Quantidade de iterações: 5

pi = 3.041839618929
```

Exemplo de execução da aplicação

```
Cálculo do valor aproximado de pi
-----------------------------------
Quantidade de iterações: 10

pi = 3.091623806668
```

Exemplo de execução da aplicação

```
Cálculo do valor aproximado de pi
-----------------------------------
quantidade de iterações: 100

pi = 3.136592684839
```

Solução:

```
clc; clear;
printf("Cálculo do valor aproximado de pi\n");
printf("-----------------------------------\n");

n = input("Quantidade de iterações: ");
while n < 1
    printf("  Valor inválido. Tente novamente.\n\n");
    n = input("Quantidade de iterações: ");
end

soma = 0;
x = 1;
i = 0;
while i < n
    soma = soma + 1 / (x * (x+2));
    x = x + 4;
    i = i + 1;
end

piaprox = 8*soma;
printf("\npi = %.12f\n", piaprox);
```

Tarefa 3: Aplicações financeiras

Em uma empresa trabalham dois amigos, Charlie e Alan. Eles são ambiciosos e desejam economizar para futuramente abrirem sua própria empresa. Todo mês Charlie aplicará 80% do seu salário na caderna de poupança, que está rendendo 2% ao mês, e Alan aplicará 50% do seu salário no fundo de renda fixa, que está rendendo 5% ao mês.

Escreva um programa que receba os salários de Charlie e de Alan, e calcula e mostra a quantidade de meses necessários para que o valor da aplicação de pertencente a Alan ultrapasse o valor da aplicação pertencente a Charlie.

Exemplo de execução da aplicação

```
Comparação de duas aplicações
=====================================
Informe o salário de Charlie ...: 3000
Informe o salário de Alan ......: 2000
Após 49 meses de aplicação:
Rendimentos de Charlie: 196657.42
Rendimentos de Alan: 198426.66
```

Solução:

```
clc;
clear;

printf("Comparação de duas aplicações\n");
printf("=====================================\n");

sal_charlie = input("Informe o salário de Charlie ...: ");
sal_alan = input("Informe o salário de Alan ......: ");

apl_charlie = 0;
apl_alan = 0;

meses = 0
while apl_alan <= apl_charlie
    meses = meses + 1;
    apl_charlie = apl_charlie * 1.02 + 0.80 * sal_charlie;
    apl_alan = apl_alan * 1.05 + 0.50 * sal_alan;
end

printf("Após %g meses de aplicação:\n", meses);
printf("Rendimentos de Charlie: %.2f\n", apl_charlie);
printf("Rendimentos de Alan: %.2f\n", apl_alan);
```