Roteiro 4 - Estruturas de Repetição

Certos tipos de problemas podem ser resolvidos com sequências de instruções executadas apenas uma vez, porém outros algoritmos requerem a execução de determinados trechos de código várias vezes, por isso o surgimento das estruturas de repetição.

Uma estrutura de repetição permite que uma sequência de instruções seja executada várias vezes até que uma condição seja satisfeita. Por isso, para saber quando utilizar é necessário analisar se uma mesma sequência de instruções necessita ser executada várias vezes no programa.

Exemplo: Imagine por exemplo um algoritmo que tenha que escrever os 1000 primeiros números positivos sem utilizar estrutura de repetição.

```
1   #include <iostream>
2
3   using namespace std;
4
5   int main()
6   {
7       cout << 1 << endl;
8       cout << 2 << endl;
9       cout << 3 << endl;
10      ...
11      cout << 1000 << endl;
12      return 0;
13  }
14
```

O código exemplificado acima é uma solução não prática para resolução do problema de imprimir os 1000 primeiros números positivos, percebe-se a repetição da tarefa de impressão dos números. Nesse caso é viável a utilização de estruturas de repetição.

Estrutura de repetição while

O **while** é uma estrutura de repetição que pode ser utilizada quando o número de repetições necessárias não é fixo. Os comandos serão repetidos até uma **condição** assumir o valor **falso**. Nesse tipo de estrutura, o **teste condicional** ocorre no **início** do bloco de comandos. Isto significa que existe a possibilidade da repetição **não ser executada**, ou seja, quando a condição assumir valor falso logo na primeira verificação. A sintaxe geral do comando é dada como se segue:

```
while (condição)
    {
      Bloco de comandos;
    }
```

Estrutura de repetição do - while

O **do-while** é uma estrutura de repetição que pode ser utilizada quando o número de repetições necessárias não é fixo. Os comandos serão repetidos até uma **condição** assumir o valor **falso**. Nesse tipo de estrutura, o **teste condicional** ocorre no **fim** do bloco de comandos. Isto significa que o bloco de comandos da repetição será executado pelo menos uma vez, ou seja, quando o bloco for executado e, ao final, a condição assumir o valor falso. A sintaxe geral do comando é dada como se segue:

```
do{
        Bloco de comandos;
  } while (condição);
```

Estrutura de repetição for

Essa estrutura de repetição é utilizada quando se conhece o número de vezes que o bloco de comandos deve ser repetido, ou seja, o número de repetições é definido. A sintaxe geral do comando **for** é composta por 3 fases: inicialização da variável de controle; teste da condição de parada; atualização ou incremento da variável de controle.

```
for (i = valor inicial; condição; incremento ou decremento de i)
     {
       Bloco de comandos;
     }
```

Exemplos

Exemplo 1: Digite e compile o exemplo abaixo que imprime uma sequência de números.

```
#include <iostream>

using namespace std;

int main()
{
    int x, y;
    x = 1;
    y = 5;
    while(x<y){
        x += 2;
        y += 1;
        cout<<"\n O valor de x e : "<<x<<endl;
        cout<<"\n O valor de y e : "<<y<<endl;
    }
    return 0;
}
```

Faça testes com o código do exemplo acima (mude valores dos incrementos de x e y; altere o teste condicional; retire os comandos de impressão para fora do bloco de repetições) e veja o que acontece.

Exemplo 2: O exemplo 2 é similar ao exemplo 1, porém utiliza a estrutura de repetição **do-while**.

```
#include <iostream>

using namespace std;

int main()
{
    int x, y;
    x = 1;
    y = 5;
    do{
        x += 2;
        y += 1;
        cout<<"\n O valor de x e : "<<x<<endl;
        cout<<"\n O valor de y e : "<<y<<endl;
    }while(x<y);
    return 0;
}
```

Responda: Os códigos sempre retornam o mesmo resultado? Justifique.
Faça os mesmos testes do exemplo 1.

Exemplo 3: Digite e compile o código abaixo que tem o objetivo de receber **N** valores digitados pelo usuário, imprimindo ao final a quantidade de valores digitados e a soma dos mesmos.

```cpp
#include <iostream>

using namespace std;

int main()
{
    int N, valor, i, soma = 0;
    cout<<"\n Digite o numero de valores a serem digitados "<<endl;
    cin>>N;
    for(i=1; i<= N; i++){
        cout<<"\n Digite o valor do numero referente a repeticao "<<i<<": "<<endl;
        cin>>valor;
        soma += valor;
    }
    cout<<"\n Foram digitados "<<N<<" valores!"<<endl;
    cout<<"\n A soma dos valores digitados e: "<<soma<<endl;
    return 0;
}

```

Faça testes com o código do exemplo acima (retire a inicialização da variável soma = 0; coloque os comandos de impressão para dentro do bloco de repetições) e veja o que acontece.

Exercícios

1. **Pesquisa sobre capacidade de armazenamento de cada tipo básico de variáveis (int, float, double, long). Modificadores de tipos de dados. Conversão implícita e explicita de tipos de dados.**

2. Escreva um programa que imprima os números pares de 0 à 50. Utilize a estrutura **for.**

3. Escreva um algoritmo que imprima os números de 100 à 1 **(ordem decrescente).**

4. Faça um programa que receba dez números do usuário e imprima a soma dos quadrados de cada número.

5. Faça um programa que leia a nota e o nome de 10 alunos na prova de algoritmos e imprima a maior e a menor nota computada e qual aluno tirou tais notas. Além disso, calcule e imprima também a soma e a média de todas as notas.

6. Desenvolva um algoritmo que recebe 10 números e imprima a quantidade de números pares e a soma de números ímpares digitados pelo usuário.

7. No último ano foi realizada um estudo estatístico sobre acidentes de trânsito em 5 cidades brasileiras. Para isso os seguintes dados foram coletados:
    a) Nome da cidade (String ou char. Necessário pesquisar sobre as funções da biblioteca string)
    b) Número de veículos
    c) Número de acidentes de trânsito

Com esses dados deseja-se saber:

a) O maior e o menor índice de acidentes e o nome da cidade a que pertencem
b) A razão entre quantidade de acidentes por quantidade de veículos nas 5 cidades analisadas
c) A média de veículos nas cinco cidades
d) A média de acidentes de trânsito nas cidades com menos de 200 veículos

8. Faça um algoritmo para identificar se um determinado número fornecido pelo usuário é primo ou não. Lembrando que um número primo só é divisível por 1 e por ele mesmo. Exemplos: 2, 5, 7, etc.

9. Faça um programa em C++ que imprima os **N** primeiros termos da serie de Fibonacci. Sabe-se que N é fornecido pelo usuário. Fibonacci = 1, 1, 2, 3, 5, 8, 13, 21, ...

10. A série de Ricci se difere da serie de Fibonacci somente pelo fato dos dois primeiros números serem fornecidos pelo usuário. Faça um programa que imprima os **N** primeiros termos da série de Ricci.

11. Elabore um algoritmo que faça a conversão de um número binário de **N** bits digitados pelo usuário para o número na base octal, base decimal e base hexadecimal. **Valide as entradas** (ou seja, só devem ser inserido bits na entrada de dados).

12. Escreva um algoritmo que calcule o m.d.c. (máximo divisor comum) entre A e B (número inteiros e positivos). Esses dois valores são passados pelo usuário através do teclado. **Utilize a lógica do algoritmo de Euclides.**

13. Faça um algoritmo que imprima a soma da sequência apresentada: H = 1 + 1/2 + 1/3 + ... + 1/N .

14. Faça um algoritmo que imprima a soma da sequência apresentada: H = 1 - 1/2 + 1/3 – 1/4 + 1/5... 1/N .

15. Faça um algoritmo que simule o funcionamento de uma calculadora que contenha as operações aritméticas básicas com dois números digitados pelo usuário. O programa implementado deve mostrar seguinte menu ao usuário. Não se esqueça de verificar se as operações podem ser realizadas.

```
====================
Calculadora de Fulano
====================
 Opções:
        1 - Soma
        2 - Subtração
        3 - Multiplicação
        4 - Divisão
        5 – Sair
====================
```

16. Escrever um algoritmo que leia um número **N** que indica quantos valores devem ser lidos. A seguir, para cada número lido, mostre uma tabela contendo o valor lido e o fatorial deste valor.

17. Faça um programa que deve solicitar números para o usuário até que seja digitado -1. Quando o usuário digitar -1, o programa termina e imprime a média de todos os números positivos digitados.
18. Faça um programa que imprima os caracteres da tabela ASCII de códigos 32 à 255. O programa deve imprimir cada carácter, seu código decimal e seu código hexadecimal. Obs.: Utilize os manipuladores de bases numéricas.

19. Construa um algoritmo para calcular a média de valores PARES e ÍMPARES, que serão digitados pelo usuário. Ao final o algoritmo deve mostrar estas duas médias bem como o maior número PAR e o menor número ÍMPAR digitado. O algoritmo finaliza quando o usuário digitar um valor negativo.

20. Implemente um programa que receba de entrada um número inteiro qualquer, após isso verifique se o número inserido é ou não um PALÍNDROMO, ou seja, o número é o mesmo tanto de visto da direita para esquerda quanto da esquerda para a direita. Ex: 121, 1441, 34643, etc...

21. Sabe-se que um país A possui 500000 habitantes e uma taxa de natalidade de 3% ao ano, já o país B possui 700000 habitantes e uma taxa de natalidade de 2% ao ano. Escreva um algoritmo, sabendo que estamos no ano de 2015, que calcule em que ano a população do país A ultrapassará a população de B.

22. Considere uma linha ferroviária entre São Paulo e Curitiba. Suponha que uma locomotiva (trem) A parte de São Paulo para Curitiba com velocidade de 30 m/s enquanto que uma outra locomotiva B parte de Curitiba para São Paulo no mesmo instante com velocidade de 40 m/s. Considere a distância entre São Paulo e Curitiba de 400 Km. Implemente um algoritmo em que calcule iterativamente o tempo necessário para os maquinistas pararem as locomotivas antes que uma colisão aconteça. O algoritmo deve calcular também a distância que as locomotivas devem percorrer para que a colisão aconteça.

23. Uma Empresa de fornecimento de energia elétrica faz a leitura mensal dos medidores de consumo. Para cada consumidor, são digitados os seguintes dados:
    - Número do consumidor;
    - Quantidade de kWh consumidos durante o mês;
    - Tipo (código) do consumidor.
        1 – residencial, preço em reais por kWh = 0,3
        2 – comercial, preço em reais por kWh = 0,5
        3 – industrial, preço em reais por kWh = 0,7

    Os dados devem ser lidos até que seja encontrado um consumidor com Número 0 (zero). Escreva um programa que calcule e imprima:
    - O custo total para cada consumidor;
    - O total de consumo para os três tipos de consumidor;
    - A média de consumo dos tipos 1 e 2.

24. O valor aproximado de PI pode ser calculado usando os 51 primeiros termos da seguinte série:

$$S = 1 - \frac{1}{3^3} + \frac{1}{5^3} - \frac{1}{7^3} + \frac{1}{9^3} \cdots$$

Sendo PI = $\sqrt[3]{S \times 32}$. Sabendo disso, implemente um algoritmo que calcule e imprima o valor de PI utilizando a série apresentada.

25. Implementar um algoritmo para calcular o sen(X). Sabe-se que o valor de X deverá ser fornecido pelo usuário em graus. O valor do seno de X será calculado pela soma dos **15 primeiros termos** da série a seguir:

$$\text{sen}(X) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \ldots$$

26. Escreva um programa em C++ que imprima o triângulo abaixo, em que a altura do triângulo (número de linhas) é fornecido pelo usuário.

```
#
# #
# # #
# # # #
# # # # #
# # # # # #
# # # # # # #
```

27. Implemente um programa que solicite um valor inteiro positivo (**N**) ao usuário. Após isso imprima a tabuada dos números de 1 à N.

28. Escrever um algoritmo que lê um número não determinado de valores para **M**, todos inteiros e positivos, um de cada vez. Se **M** for par, verificar quantos divisores possui e escrever esta informação. Se **M** for ímpar e menor do que 10 calcular e escrever o fatorial de **M**. Se **M** for ímpar e maior ou igual a 10 calcular e escrever a soma dos inteiros de 1 até **M**. O algoritmo deve finalizar quando for digitado o valor zero para **M.**

29. Suponha que C++ possua somente as operações de soma e subtração. Dados dois números inteiros positivos A e B, determine o quociente e o resto da divisão de A por B.

30. Faça um programa que calcule o número de combinações e arranjos de **n** objetos diferentes, onde **p** objetos são escolhidos de cada vez. Ao valores de **n** e **p** devem ser fornecidos pelo usuário. As fórmulas que calculam são dadas por:

Cn_p = n!/ (p! * (n - p)!) **An_p = n!/(n-p)!**

31. Implemente um programa que solicite duas letras de ‘a’ à ‘z’ e imprima o número de caracteres que estão entre eles. Exemplo: Caso usuário tenha digitado ‘c’ e ‘g’, deve-se retornar que existem 3 caracteres entre eles.

32. Faça um programa que receba de entrada o sexo, altura e peso de **N** pessoas. Determine a soma dos pesos das mulheres com mais de 30 anos e a altura do homem mais alto.

33. A companhia de teatro do IFMG Sabará deseja realizar uma série de espetáculos. A direção calcula que a despesa fixa do espetáculo é de R$200,00. Além disso, sabe-se que com os ingressos ao preço de R$5,00 serão vendidos 120 entradas. Em uma pesquisa de público estimou-se que a cada R$0,50 de diminuição no valor do ingresso espera-se um aumento de 26 ingressos nas vendas. Diante dessas informações, implemente um programa que calcule e imprima uma tabela contendo os valores dos lucros esperados em função do valor dos ingressos, fazendo uma variação nos ingressos de R$5,00 à R$1,00 de R$0,50 em R$0,50. Escreva ainda o lucro máximo esperado, o preço do ingresso e quantidade de ingressos vendidos para obtenção desse lucro máximo estimado.

34. Em uma eleição presidencial existem quatro candidatos. Os votos são informados através de códigos. Os dados utilizados para a contagem dos votos obedecem à seguinte codificação:
    - 1, 2, 3, 4: voto para os respectivos candidatos;
    - 5: voto em branco;
    - Outros valores: voto nulo.

    Elabore um algoritmo que leia o código do candidato em um voto. Calcule e escreva:
    - total de votos para cada candidato;
    - total de votos nulos;
    - total de votos em branco;

    Como finalizador do conjunto de votos, tem-se o valor de código igual à 0.

35. Escrever um algoritmo que gera e escreve os 5 primeiros números perfeitos. Um número perfeito é aquele que é igual a soma dos seus divisores. (Ex.: 6 = 1+2+3; 28= 1+2+4+7+14 etc).

36. O número de inscrição no CPF é composto de onze dígitos decimais, sendo os oito primeiros aleatoriamente designados no momento da inscrição. Já o nono (antepenúltimo) dígito indica a região fiscal responsável pela inscrição (MG é a região 6, portanto em todos CPF's emitidos em MG o nono digito é 6). Por fim, o décimo e o décimo primeiro são dígitos verificadores calculados de acordo com um algoritmo definido pela Receita Federal e publicamente conhecido. Assim sendo, seja D os nove primeiros dígitos de um número de CPF qualquer visitado da direita para a esquerda, na forma:
    D = (d0, d1, d2, d3, d4, d5, d6, d7, d8), onde di representa o dígito do CPF na posição i e d0 representa a posição mais a direita deste CPF. Ou seja, os dígitos do CPF 123456789 são representados por D = (9, 8, 7, 6, 5, 4, 3, 2, 1). Diante disso, os dígitos verificadores v1 e v2 podem ser calculados pelas expressões:

    Assim, implemente um algoritmo que receba de entrada o valor dos 9 primeiros dígitos de um CPF (uma única variável inteira), e dos 2 dígitos verificadores. Desmembre os 9 dígitos do CPF utilizando as operações de divisão e resto da divisão inteira. Faça os cálculos, de acordo com a expressão acima, para verificar se os dígitos verificadores do CPF digitado são válidos. **Utilizem estrutura de repetição para calcular os dígitos verificadores.** Como exemplo, utilizem o valor inicial dos CPF 111444777, os dígitos verificadores devem ser v1 = 3 e v2 = 5. Ou utilizem o próprio CPF como exemplo.