



Algoritmos - Estrutura de Repetição - While - Básico

Tarefa - Algoritmo - Estrutura de Repetição - While - Básico

1 Estrutura de Repetição - While - Básico

O **while** da linguagem C representa uma estrutura de repetição do tipo enquanto, esta estrutura é controlada por uma expressão booleana e será executada enquanto a expressão for verdadeira.

Esta estrutura é ideal para representar uma repetição para o qual não se sabe o número de vezes que ela deverá ocorrer. O exemplo abaixo é baseado no conceito de aleatoriedade, ele utiliza a função **rand** para gerar valores aleatórios. O resto da divisão, simbolizado por %, faz com o que o valor aleatório fique na faixa de valores que inicia em zero e termina em 999. Em princípio, não é possível determinar quantas vezes a função **printf** interna ao **while** será executada (será que isso é realmente verdade?).

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main()
6 {
7     srand(time(0));
8     int aleatorio = 0;
9     while ((aleatorio = random() % 1000) < 500)
10         printf("aleatório: %d\n", aleatorio);
11     printf("último aleatório gerado: %d\n", aleatorio);
12     return 0;
13 }
```

Programa 1: Exemplo de programa utilizando a estrutura de repetição **for**

O programa de exemplo também tem uma característica peculiar, ele faz a atribuição e teste utilizando apenas uma linha de código. Repare que é utilizado um operador de atribuição dentro do parenteses mais interno, isso fará com que o valor aleatório gerado pela função **rand** e submetido ao resta da divisão por 100 seja armazenado na variável **aleatorio**, em seguida um teste para saber se o número é menor do que 500 é realizado, o resultado do teste gerará um valor booleano que será utilizado pelo comando **while** para testar se deve executar a ação ou sair da repetição.

Estruturas de repetição do tipo enquanto são úteis, por exemplo, para verificar valores que são fornecidos pelo usuário, ele poderá tentar fornecer uma entrada inválida e repetir a entrada

inválida por diversas vezes, nessa situação o teste da repetição ficará como o valor verdade verdadeiro e será repetido até que a entrada seja valida, ao se tornar válida o valor verdade do teste será falso e a repetição será finalizada.

Considere utilizar uma estrutura de repetição deste tipo para um processo na qual a quantidade de vezes não é determinada.

Existe uma variação da estrutura de repetição enquanto que é a estrutura faça enquanto, ela é representada pelos comandos **do** e **while**, nessa estrutura é possível executar um comando ou um bloco de comandos e depois verificar a condição de repetição.

Os livros [Paul Deitel, 2022] e [Brian W. Kernighan, 1988] podem ser utilizados para obter mais informações sobre a estrutura de repetição **for**.

2 Exercícios

1. Faça um programa que faça a leitura de um inteiro estritamente positivo n . Utilize uma estrutura de repetição **while** para ficar repetindo a entrada enquanto os valor fornecido pelo usuário $\text{for} \leq 0$.
2. Faça um programa para ler valores inteiros através do teclado. O programa deverá exibir par ou impar para números positivos e encerrar quando algum ler um número ≤ 0 .
3. Faça um programa para gerar uma quantidade n de números aleatórios diferentes entre dois valores um valor inicial i e um valor final f .
4. Estude as funções **rand**, **srand** e **time**. Por que é interessante utilizar a função **srand**?
5. Faça um programa para converter um número da base decimal (base 10) para um valor na base binária (base 2).
6. Dada um valor monetário r (sem centavos), descreva a menor quantidade de valores de notas de 100, 50, 20, 10, 5, 2 e 1 que representa essa quantia.
7. Faça um programa para determinar se um número é primo ou não. Um número primo é um número que somente é divisível por 1 e por ele mesmo.
8. Faça um programa para calcular a raiz quadrada de um certo número real x com um valor de precisão $\delta \geq 0$. Os valores x e δ deverão ser representado como números de ponto flutuante.
9. Dado um array de inteiros que esteja ordenado, verifique se um determinado valor está lá ou não e, caso estiver, informe a posição. Dado que o array está ordenado é possível utilizar uma busca binária, ou seja, comece procurando no valor que está no meio do array, se o valor estiver sendo procurado for maior procure novamente a partir da metade do array e o final, caso contrário, verifique entre o inicial e a metade do array. Repita esse procedimento sempre descartando metade do novo array formado.

10. Um experimento consiste para comparar o crescimento de duas populações. A população A crescerá com taxa t_A e iniciará com n_A indivíduos. A população B crescerá com taxa t_B e iniciará com n_B indivíduos. Considerando que a população menor terá uma taxa de crescimento maior do que a população maior, qual é o número de períodos que a população menor levará para ultrapassar a maior? Faça um programa para simular isso.

Referências

[Brian W. Kernighan, 1988] Brian W. Kernighan, D. M. R. (1988). *C Programming Language*. Prentice Hall, 2 edition.

[Paul Deitel, 2022] Paul Deitel, H. D. (2022). *C How to Program*. Pearson, 9 edition.