Material de Apoio para Maratona de Programação em C

1. Introdução

Maratonas de programação são competições desafiadoras que exigem habilidades em lógica, algoritmos e estruturas de dados. Este material foca em fornecer funções úteis e exemplos práticos para resolver problemas com a linguagem C, que é amplamente utilizada por sua eficiência e controle de baixo nível.

2. Funções Essenciais

Aqui estão algumas funções essenciais para leitura e manipulação de dados em C:

- Leitura de Inteiros, Floats e Strings.
- Saída formatada para exibição clara dos resultados.
- Funções matemáticas úteis como cálculo de MDC e MMC.

2.1 Exemplo: Leitura de Strings

```
Função para ler strings removendo o caractere '\n':
    ```c

#include <stdio.h>

void readString(char *str, int maxLength) {
 fgets(str, maxLength, stdin);
 str[strcspn(str, '\n')] = 0; // Remove o '\n'
}

Uso:
    ```c
```

Material de Apoio para Maratona de Programação em C

```
char name[50];
printf("Digite seu nome: ");
readString(name, 50);
printf("Bem-vindo, %s!\n", name);
```

3.1 Exemplo: QuickSort

```
Função de ordenação QuickSort:
```C
void quickSort(int arr[], int low, int high) {
 if (low < high) {</pre>
 int pi = partition(arr, low, high);
 quickSort(arr, low, pi - 1);
 quickSort(arr, pi + 1, high);
 }
}
int partition(int arr[], int low, int high) {
 int pivot = arr[high];
 int i = (low - 1);
 for (int j = low; j <= high - 1; j++) {
 if (arr[j] < pivot) {</pre>
 i++;
 int temp = arr[i];
 arr[i] = arr[j];
```

# Material de Apoio para Maratona de Programação em C

```
arr[j] = temp;
 }
 }
 int temp = arr[i + 1];
 arr[i + 1] = arr[high];
 arr[high] = temp;
 return (i + 1);
}
. . .
Uso:
```C
int arr[] = \{10, 7, 8, 9, 1, 5\};
int n = sizeof(arr[0]);
quickSort(arr, 0, n - 1);
for (int i = 0; i < n; i++) {
   printf("%d ", arr[i]);
}
. . .
```