

Aluna: Ana Beatriz Rocha Sousa

Respostas - Exercício 03

Exercício 03

1-A manipulação de bits envolve a alteração de bits individuais em um número binário. Isso pode ser feito por meio de várias operações, incluindo as operações lógicas AND, OR e XOR, bem como deslocamentos de bits para a esquerda e para a direita.

2-É importante para otimizar a eficiência e o desempenho do código. Ela permite que os programadores realizem operações complexas de forma rápida e eficiente, melhorando o desempenho do código e reduzindo o uso de recursos.

3-As operações básicas de manipulação de bits incluem AND bit a bit, OR bit a bit, XOR bit a bit e deslocamentos de bits para a esquerda e para a direita.

4- #include <stdio.h>

// Função para imprimir os bits de um número

```
void imprimir_bits(int num) {  
    int i;  
    for (i = 31; i >= 0; i--) {  
        (num & (1 << i)) ? printf("1") : printf("0");  
    }  
    printf("\n");  
}
```

```
int main() {  
    int num1, num2, opcao;  
  
    printf("Digite o primeiro número: ");  
    scanf("%d", &num1);  
  
    printf("Digite o segundo número: ");  
    scanf("%d", &num2);  
  
    printf("Escolha uma opção:\n");  
    printf("1 - Imprimir bits dos números\n");  
    printf("2 - Realizar operação AND\n");  
    printf("3 - Realizar operação OR\n");  
    printf("4 - Realizar operação XOR\n");  
    scanf("%d", &opcao);  
  
    switch (opcao) {  
        case 1:  
            printf("Bits do número %d: ", num1);
```

```
    imprimir_bits(num1);
    printf("Bits do número %d: ", num2);
    imprimir_bits(num2);
    break;
case 2:
    printf("Resultado da operação AND: %d\n", num1 & num2);
    imprimir_bits(num1 & num2);
    break;
case 3:
    printf("Resultado da operação OR: %d\n", num1 | num2);
    imprimir_bits(num1 | num2);
    break;
case 4:
    printf("Resultado da operação XOR: %d\n", num1 ^ num2);
    imprimir_bits(num1 ^ num2);
    break;
default:
    printf("Opção inválida\n");
}

return 0;
}
```