

# Pratica 3

## Exercicio 1:

não recursivo

```
#include <stdio.h>

int main() {
    int i;
    printf("Informe o limite: ");
    scanf("%d", &i);

    for (int num = 0; num <= i; num++) {
        int num2;
        int resto = num % 2;
        if (resto == 0) {
            num2 = -(num / 2);
        } else {
            num2 = (num / 2) + 1;
        }
        printf("[%d, %d] ", num, num2);
    }

    return 0;
}
```

recursivo

```
#include <stdio.h>

void limite(int num, int i) {
    if (num <= i) {
        int num2;
        int resto = num % 2;
        if (resto == 0) {
            num2 = -(num / 2);
        } else {
            num2 = (num / 2) + 1;
        }
        printf("[%d, %d] ", num, num2);
        limite(num + 1, i);
    }
}

int main() {
    int i;
    printf("Digite um número inteiro: ");
    scanf("%i", &i);

    if (i == 0) {
        printf("[0, 0]\n");
    } else {

```

```

        limite(0, i);
        printf("\n");
    }

    return 0;
}

```

## Exercicio 2:

```

#include <stdio.h>

int soma(int m, int n) { //n = 0, então m + n = m
    if (n == 0) {
        return m;
    }
    else {
        return soma(m + 1, n - 1); //m + s(n) = s(m + n)
    }
}

int main() {
    int num1, num2;
    printf("Primeiro número: ");
    scanf("%d", &num1);
    printf("Segundo número: ");
    scanf("%d", &num2);

    int resultado = soma(num1, num2);
    printf("%d + %d = %d\n", num1, num2, resultado);

    return 0;
}

```

## questao 3:

```

#include <stdio.h>

void menorQue(int n, int m) {
    if (m == 0) {
        return;
    }

    printf("[%d,%d] ", n, m);
    menorQue(n, m - 1);
}

void mq(int i) {
    if (i == 0) {
        printf("[0,1]\n");
        return;
    }

    mq(i - 1);
}

```

```

        for (int j = 0; j < i; j++) {
            menorQue(j, i);
            printf("; ");
        }

        printf("[%d,%d]\n", i, i + 1);
    }

int main() {
    int i;
    printf("Informe o valor de i: ");
    scanf("%d", &i);

    mq(i);

    return 0;
}

```

#### questao 4

```

#include <stdio.h>

void printmq(int n, int m) {
    for (int i = m; i > 0; i--) {
        printf("[%d,%d] ", n, i);
    }
}

void mq(int i) {
    printf("[0,1]\n");

    for (int j = 1; j <= i; j++) {
        printf("%d | ", j);
        printmq(0, j);
        for (int k = 1; k < j; k++) {
            printf("; ");
            printmq(k, j);
        }
        printf("\n");
    }
}

int main() {
    int i;
    printf("Informe o valor de i: ");
    scanf("%d", &i);

    mq(i);

    return 0;
}

```

#### questão 5

```

#include <stdio.h>

const int tamanho = 4;
int matriz[4][4];

void definirDistancia(int x, int y, int valor){
    matriz[x][y] = valor;
    matriz[y][x] = valor;
}

void estradas(int cidade1, int cidade2){
    int n = 1;
    if ((cidade1 == 0 && cidade2 == 2) || (cidade1 == 2 && cidade2 == 0)
        || (cidade1 == 1 && cidade2 == 3) || (cidade1 == 3 && cidade2 == 1))
        n = 2;
    printf("Entre a cidade %d e %d, há %d estradas.\n", cidade1, cidade2, n);
}

int distanciaRecurativa(int cidade1, int cidade2) {
    if (cidade1 == cidade2) {
        return 0;
    } else if (matriz[cidade1][cidade2] != 0) {
        return matriz[cidade1][cidade2];
    } else {
        int minDistancia = 9999; // Valor grande para comparação inicial

        for (int i = 0; i < tamanho; ++i) {
            if (matriz[cidade1][i] != 0 && i != cidade2) {
                int novaDistancia = distanciaRecurativa(i, cidade2) + matriz[cidade1][i];
                if (novaDistancia < minDistancia) {
                    minDistancia = novaDistancia;
                }
            }
        }

        return minDistancia;
    }
}

void printMatriz(){
    int i, j;
    for (i = 0; i < tamanho; i++) {
        for (j = 0; j < tamanho; j++)
            printf (" %d ", matriz[i][j]);
        printf ("\n");
    }
}

int main() {
    definirDistancia(0, 1, 20);
    definirDistancia(1, 2, 20);
    definirDistancia(1, 3, 35);
    definirDistancia(0, 2, 40);
    definirDistancia(2, 3, 15);
    definirDistancia(3, 0, 10);
    printMatriz();
}

```

```

    estradas(0, 1);
    estradas(0, 2);
    estradas(0, 3);
    estradas(1, 2);
    estradas(1, 3);
    estradas(2, 3);

    printf("A distância entre a cidade 0 e 1 é %dkm.\n", distanciaRecursiva(0, 1));
    printf("A distância entre a cidade 0 e 3 é %dkm.\n", distanciaRecursiva(0, 3));
    printf("A distância entre a cidade 1 e 2 é %dkm.\n", distanciaRecursiva(1, 2));
    printf("A distância entre a cidade 2 e 3 é %dkm.\n", distanciaRecursiva(2, 3));

    return 0;
}

```

## questao 6

```

#include <stdio.h>

int tamanho = 3;
int matriz[3][3];

void definirDistancia(int x, int y, int valor){
    matriz[x][y] = valor;
    matriz[y][x] = valor;
}

void printMatriz(){
    int i, j;
    for (i = 0; i < tamanho; i++) {
        for (j = 0; j < tamanho; j++)
            printf (" %d ", matriz[i][j]);
        printf ("\n");
    }
}

int calculaDistancia(int cidade1, int cidade2){
    int distancia;
    if (cidade1 == 0){
        if (cidade2 == 1){
            distancia = 450;
        } else if (cidade2 == 2){
            distancia = 500;
        }
    } else if (cidade1 == 1){
        if (cidade2 == 0){
            distancia = 450;
        } else if (cidade2 == 2){
            distancia = 600;
        }
    } else if (cidade1 == 2){
        if (cidade2 == 0){
            distancia = 500;
        } else if (cidade2 == 1){

```

```

        distancia = 600;
    }
}
return distancia;
}

int main(){
    int cidade1, cidade2;

    printf("Digite o número da primeira cidade: ");
    scanf("%d", &cidade1);

    printf("Digite o número da segunda cidade: ");
    scanf("%d", &cidade2);

    int distancia = calculaDistancia(cidade1, cidade2);
    printf("A distância entre a cidade %d e a cidade %d é: %d\n", cidade1, cidade2, distancia);

    printf("Matriz:\n");
    definirDistancia(0, 1, 450);
    definirDistancia(0, 2, 500);
    definirDistancia(1, 2, 600);
    printMatriz();

    printf("Há um ciclo no grafo.");
    printf("\nNo 0: grau de entrada = 0 \n      grau de saída = 2\n");
    printf("No 1: grau de entrada = 2 \n      grau de saída = 0\n");
    printf("No 2: grau de entrada = 1 \n      grau de saída = 1\n");

    return 0;
}

```