

Tópicos especiais em Engenharia de Software – 2019

Exercício sobre teste estrutural – fluxo de controle

Prof. Marcio Delamaro
PAE: Stevão Andrade

Nome e NUSP: _____

Para cada um dos programas abaixo, construa o GFC

```
1)
int particao(int vetor[], int ini, int fim)
{
    int pivo, i, j, temp;
    pivo = vetor[fim];
    i = ini; j = fim - 1;

    while (i <= j)
    {
        if (vetor[i] <= pivo) i++;
        else
            if (vetor[j] > pivo) j--;
            else
            {
                temp = vetor[i];
                vetor[i] = vetor[j];
                vetor[j] = temp;
                i++;
                j--;
            }
    }
    vetor[fim] = vetor[i];
    vetor[i] = pivo;
    return i;
}
```

```

2)
void mult(int[][]mat1, int[][] mat2)
int[][] mat3;
    mat3 = new int[mat1.length][mat2[0].length];

    //calcula o produto mat1*mat2 e armazena em mat3
    for(int linha=0; linha<mat1.length; linha++)
        for(int coluna=0; coluna<mat2[0].length; coluna++) {
            int acumula_somaprod=0;
            for(int i=0; i< mat1[0].length; i++)
                acumula_somaprod=acumula_somaprod+mat1[linha][i]*mat2[i]
[coluna];
            mat3[linha][coluna]=acumula_somaprod;
        }

    //imprime mat3
    for(int linha=0; linha<mat3.length; linha++){
        for(int coluna=0; coluna<mat3[0].length; coluna++)
            System.out.print(mat3[linha][coluna]+" ");
        System.out.println();
    }
}

```

3)

Para o programa abaixo, mostre o GFC e projete:

- a) um conjunto de teste que seja todos-nós adequado mas que não seja todas-arestas adequado;
- b) um conjunto de teste que seja todas-arestas adequado.

Para cada de teste, mostre o caminho completo executado e os requisitos que foram cobertos.

```
int daysIn[] = {0, 31, 0, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

/*****
// Calculate the number of Days between the two given days in
// the same year.
// preconditions : day1 and day2 must be in same year
//                1 <= month1, month2 <= 12
//                1 <= day1, day2 <= 31
//                month1 <= month2
//                The range for year: 1 ... 10000
//*****/

int cal (int month1, int day1, int month2,
        int day2, int year)
{
    int numDays, i;

    if (month2 == month1) /* in the same month */
        numDays = day2 - day1;
    else
    {
        /* Are we in a leap year? */
        int m4 = year % 4;
        int m100 = year % 100;
        int m400 = year % 400;
        if ((m4 != 0) || ((m100 == 0) && (m400 != 0)))
            daysIn[2] = 28;
        else
            daysIn[2] = 29;

        /* start with days in the two months */
        numDays = day2 + (daysIn[month1] - day1);

        /* add the days in the intervening months */
        for ( i = month1 + 1; i <= month2-1; i++)
            numDays = daysIn[i] + numDays;
    }
    return (numDays);
}
```