

Gabarito da Prova 2
MAC115 - Física, segundo semestre de 2002

1. A saída é como segue:

```
4 5
5 6
4
```

2. (a)

```
int pertence(double x, double B[], int m)
{
    int i;

    for (i = 0; i < m; i++)
        if (x == B[i])
            return 1;

    return 0;
}
```

```

(b) int contido(double A[], double B[], int m, int n)
    {
        int i;

        for (i = 0; i < m; i++)
            if (!pertence(A[i], B, n))
                return 0;

        return 1;
    }

(c) #include <stdio.h>
    #define NMAX 100 /* Tamanho maximo dos conjuntos */

    int pertence(double x, double B[], int m);
    int contido(double A[], double B[], int n, int m);

    int main()
    {
        int i, m, n;
        double A[NMAX], B[NMAX];

        scanf("%d", &m);
        for (i = 0; i < m; i++)
            scanf("%lf", &A[i]);
        scanf("%d", &n);
        for (i = 0; i < n; i++)
            scanf("%lf", &B[i]);

        if (n != m) {
            printf("Conjuntos diferentes\n");
            return 0;
        }
        if (contido(A, B, m, n)) { /* Como m = n, isso basta */
            printf("Conjuntos iguais\n");
            return 0;
        }

        return 0;
    }

```

```

3. (a) double vmed(double A[][NMAX], int m, int n, int i, int j)
    {
        if ((i == 0) && (j == 0))
            return (A[1][0] + A[0][1]) / 2;
        if ((i == m - 1) && (j == 0))
            return (A[m - 2][0] + A[m - 1][1]) / 2;
        if ((i == 0) && (j == n - 1))
            return (A[0][n - 2] + A[1][n - 1]) / 2;
        if ((i == m - 1) && (j == n - 1))
            return (A[m - 1][n - 2] + A[m - 2][n - 1]) / 2;

        /* supomos n, m>=2 */
        if (j == 0)
            return (A[i - 1][0] + A[i][1] + A[i + 1][0]) / 3;
        if (j == m - 1)
            return (A[i - 1][m - 1] + A[i][m - 2] + A[i + 1][m - 1]) / 3;
        if (i == 0)
            return (A[0][j - 1] + A[1][j] + A[0][j + 1]) / 3;
        if (i == m - 1)
            return (A[m - 1][j - 1] + A[m - 2][j] + A[m - 1][j + 1]) / 3;
        return (A[i - 1][j] + A[i + 1][j] + A[j - 1][i] + A[j + 1][i]) / 4;
    }

(b) void med(double A[][NMAX], int m, int n, double M[][NMAX])
    {
        int i, j;
        for (i = 0; i < m; i++)
            for (j = 0; j < n; j++)
                M[i][j] = vmed(A, m, n, i, j);
    }

```

```

(c) #include <stdio.h>

#define NMAX 100
#define TRUE 1

double vmed(double A[][NMAX], int m, int n, int i, int j);
void med(double A[][NMAX], int m, int n, double M[][NMAX]);
void copie(double A[][NMAX], int m, int n, double B[][NMAX]);

int main()
{
    int m, n, i, j, k;
    double A[NMAX][NMAX], B[NMAX][NMAX];
    scanf("%d %d %d", &k, &m, &n);
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%lf", &A[i][j]);
    while (TRUE) {
        putchar('\n');
        for (i = 0; i < m; i++) {
            for (j = 0; j < n; j++)
                printf("%4.2g ", A[i][j]);
            putchar('\n');
        }
        if (--k < 0)
            break;
        med(A, m, n, B);
        copie(A, m, n, B);
    }
    return 0;
}

void copie(double A[][NMAX], int m, int n, double B[][NMAX])
{
    int i, j;
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            A[i][j] = B[i][j];
}

```