



```
int* multmat(int* vma, int* vmb, int n, int m, int p, int q) {
    int i, s, k, h, t;

    if (vma == NULL || vmb == NULL) {
        Return NULL;
    }
    if (n <= 0 || m <= 0 || p <= 0 || q <= 0) {
        Return NULL;
    }
    if (m == p) {
        int* matrizAux = (int*) malloc(sizeof(int)*m);
        if (matrizAux == NULL) {
            Return NULL;
        }
        for (i=0; i < n; i++) {
            for (s=0; s < m; s++) {
                l = i*m + s;
                matrizAux[l] = 0;
                for (k=0; k < m; k++) {
                    t = k*m + s;
                    h = i*m + k;
                    matrizAux[l] += vma[h] * vmb[t];
                }
            }
        }
        Return matrizAux;
    }
    Return NULL;
}
```



```
int* transposta de mat (int* vma, int n, int m) {
```

```
    int i, j, k = 0;
```

```
    if (n <= 0 || m <= 0) {
```

```
        Return NULL;
```

```
    }
```

```
    int aux, trans;
```

```
    for (i = 0; i < n; i++) {
```

```
        for (j = 0; j < m; j++) {
```

```
            k = i * m + j;
```

```
            trans = j * m + i;
```

```
            if (i > j) {
```

```
                aux = vma[k];
```

```
                vma[k] = vma[trans];
```

```
                vma[trans] = aux;
```

```
            }
```

```
        }
```

```
    }
```

```
    Return NULL;
```

```
}
```



D	S	T	Q	Q	S	S
0	1	2	3	4	5	6

```
int somaDaDiagonal(int* vma, int n, int m, int l) {
```

```
    int i, s, k;
```

```
    int somaElements = 0;
```

```
    if (n <= 0 || m <= 0 || l <= 0) {
```

```
        Return NULL;
```

```
    }
```

```
    int* elems = (int*) malloc(sizeof(int) * m);
```

```
    if (elems == NULL) {
```

```
        Return NULL;
```

```
    }
```

```
    for (i = 0; i < m; i++) {
```

```
        k = l * m + i;
```

```
        elems[i] = vma[k];
```

```
    }
```

```
    for (i = 0; i < m; i++) {
```

```
        somaElements += elems[i];
```

```
    }
```

```
    Return somaElements;
```

```
}
```



```
int ESimetria (int *vma, int n, int m) {
```

```
    int i, j, k, h;
```

```
    if (n <= 0 || m <= 0) {
```

```
        Return NULL;
```

```
    }
```

```
    for (i = 0; i < n; i++) {
```

```
        for (j = i + 1; j < m; j++) {
```

```
            k = i * m + j;
```

```
            h = j * m + i;
```

```
            if (vma[k] != vma[h]) {
```

```
                Return 0;
```

```
            }
```

```
        }
```

```
    }
```

```
    Return 1;
```

```
}
```



int* TRaInguloSuperiorde mat (int* vma, int n, int m) {

int i, s, k;

if (n <= 0 || m <= 0) {

Return NULL;

}

int* vetor = (int*) malloc (sizeof(int) * ((n*m)-n)/2);

if (vetor == NULL) {

Return NULL;

}

for (i=0; i < n; i++) {

for (s=0; s < m; s++) {

if (i < s) {

k = i*m + s;

vetor[i] = vma[k];

}

}

}

Return vetor;

}



```
int ElementosDiagonal(int* vma, int n, int m) {
```

```
    int i, j, k;
```

```
    int elementos = 0;
```

```
    if (n <= 0 || m <= 0) {
```

```
        Return Return NULL;
```

```
    }
```

```
    for (i = 0; i < n; i++) {
```

```
        for (j = 0; j < m; j++) {
```

```
            if (i == j) {
```

```
                k = i * m + j;
```

```
                elementos += vma[k];
```

```
            }
```

```
        }
```

```
    }
```

```
    Return elementos;
```

```
}
```