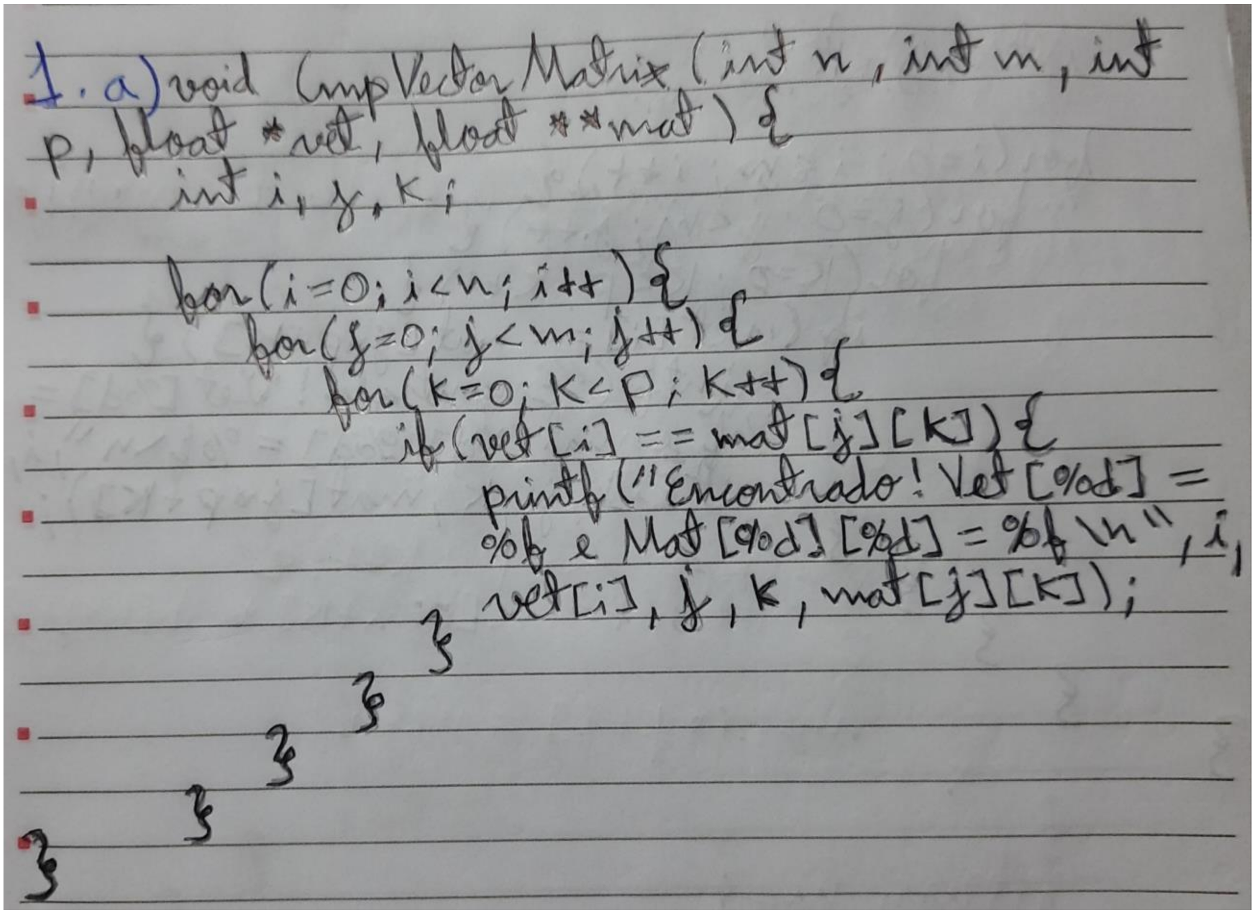
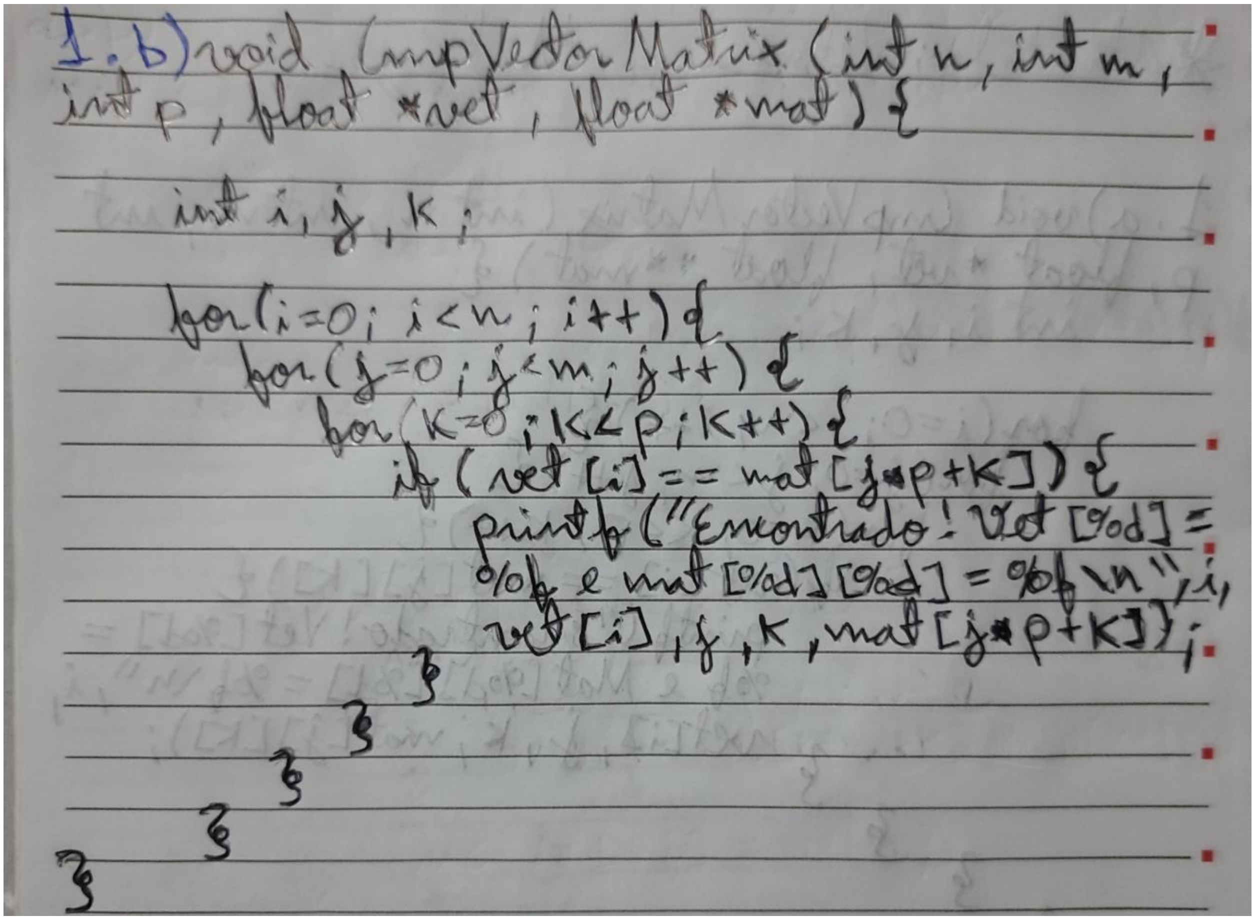
Lista de Exercicios 01 – Vetores e Matrizes

Aluno: Daniel Sousa Goncalves

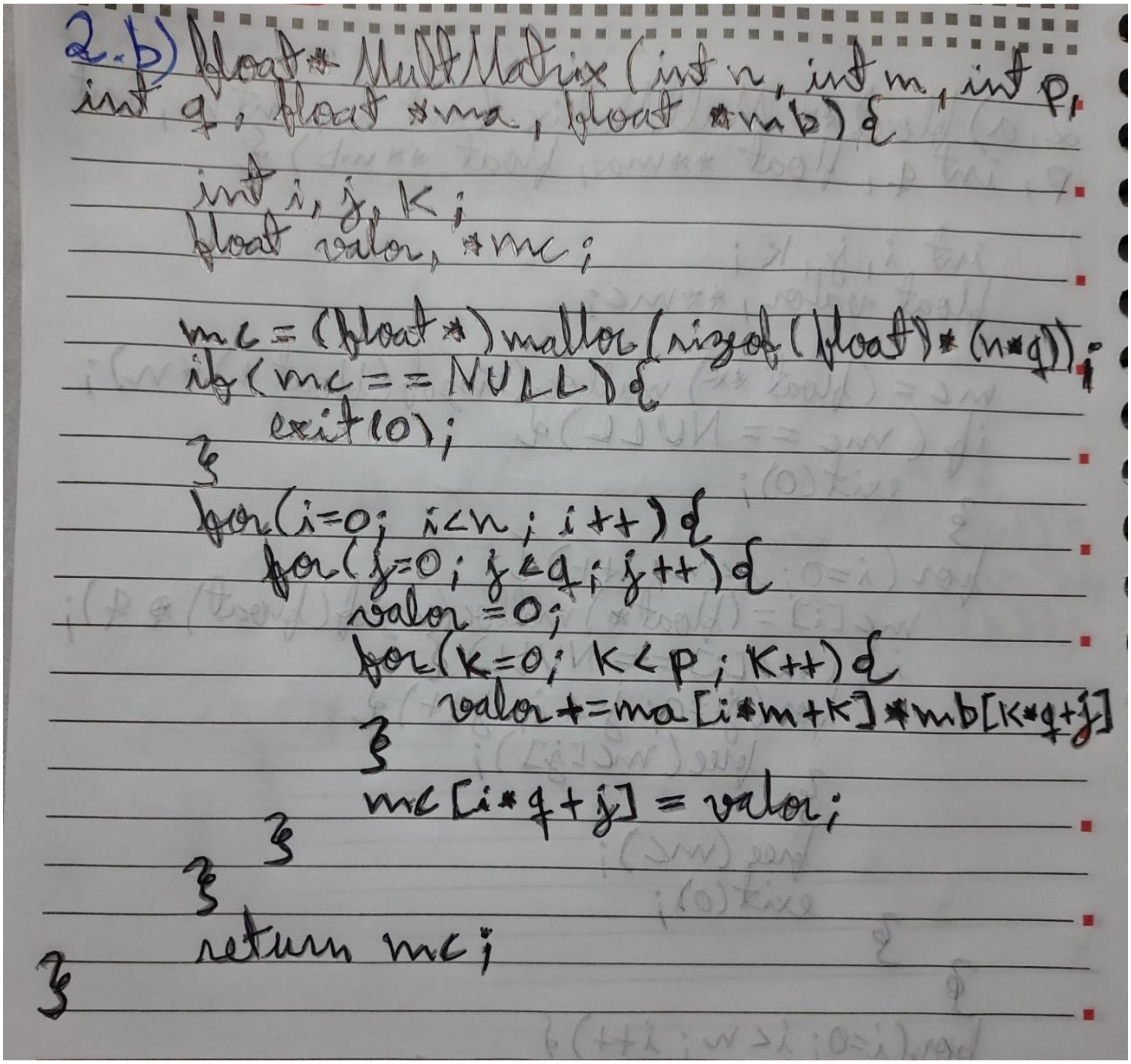


1. a)
```
void CmpVectorMatrix (int n, int m, int
p, float *vet, float **mat) {
    int i, j, k;

    for(i=0; i<n; i++) {
        for(j=0; j<m; j++) {
            for(k=0; k<p; k++) {
                if(vet[i] == mat[j][k]) {
                    printf("Encontrado! Vet[%d] =
                    %f e Mat[%d][%d] = %f \n", i,
                    vet[i], j, k, mat[j][k]);
                }
            }
        }
    }
}
```



```
1.b) void CmpVetorMatrix (int n, int m,
int p, float *vet, float *mat) {

    int i, j, k;

    for(i=0; i<n; i++) {
        for(j=0; j<m; j++) {
            for(k=0; k<p; k++) {
                if (vet[i] == mat[j*p+k]) {
                    printf("Encontrado! Vet[%d] =
                    %f e mat[%d][%d] = %f \n", i,
                    vet[i], j, k, mat[j*p+k]);
                }
            }
        }
    }
}
```

```
2.a) float** MultMatix (int n, int m, int
p, int q, float **ma, float **mb) {

    int i, j, k;
    float valor, **mc;

    mc = (float **) malloc(sizeof(float *) * n);
    if (mc == NULL){
        exit(0);
    }
    for (i=0; i<n; i++){
        mc[i] = (float*) malloc(sizeof(float) * q);
        if (mc[i] == NULL) {
            for(j=0; j < i; j++) {
                free(mc[j]);
            }
            free (mc);
            exit(0);
        }
    }
    for(i=0; i<n; i++){
        for(j=0; j<q; j++){
            valor = 0;
            for(k=0; k<m; k++){
                valor += ma[i][k] * mb[k][j];
            }
            mc[i][j] = valor;
        }
    }
    return mc;
}
```



```
2.b) float* MultMatrix (int n, int m, int p,
int q, float *ma, float *mb){

    int i, j, K;
    float valor, *mc;

    mc = (float*)malloc(sizeof(float)*(n*q));
    if(mc == NULL){
        exit(0);
    }
    for(i=0; i<n; i++){
        for(j=0; j<q; j++){
            valor =0;
            for(K=0; K<p; K++){
                valor+=ma[i*m+K]*mb[K*q+j]
            }
            mc[i*q+j] = valor;
        }
    }
    return mc;
}
```