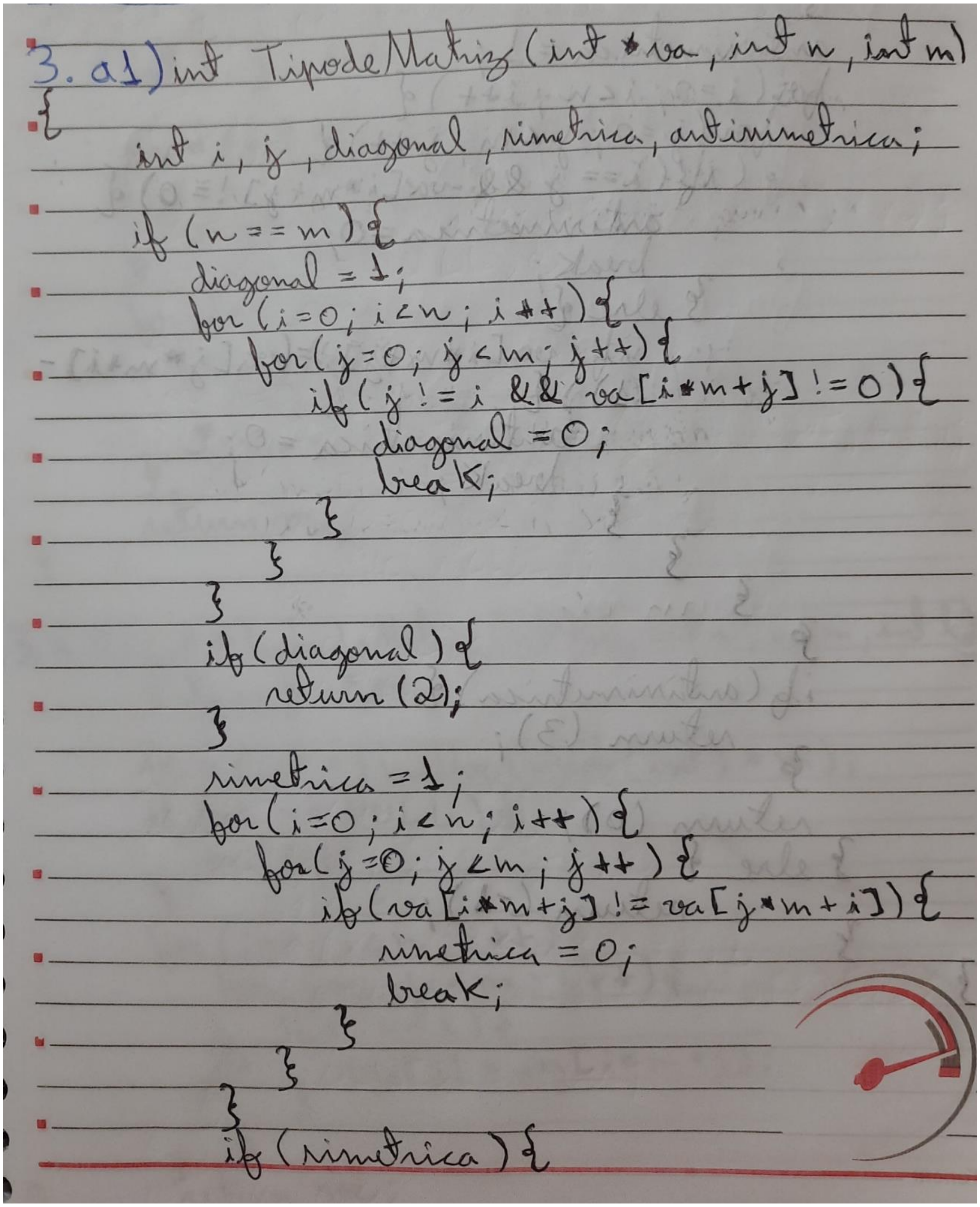
Lista de Exercicios 02 – Vetores e Matrizes

Aluno: Daniel Sousa Goncalves



```
3. a1) int TipodeMatriz(int *va, int n, int m)
{
     int i, j, diagonal, simetrica, antisimetrica;

     if (n == m){
        diagonal = 1;
        for(i=0; i<n; i++){
           for(j=0; j<m; j++){
              if(j != i && va[i*m+j] != 0){
                 diagonal = 0;
                 break;
              }
           }
        }
        if(diagonal){
           return (2);
        }
        simetrica = 1;
        for(i=0; i<n; i++){
           for(j=0; j<m; j++){
              if(va[i*m+j] != va[j*m+i]){
                 simetrica = 0;
                 break;
              }
           }
        }
        if(simetrica){
```

```
        return (1);
    }
    antisimetrica = 1;
    for(i=0; i<n; i++){
        for(j=0; j<m; j++){
            if(i== j && va[i*m+j] != 0){
                antisimetrica =0;
                break;
            } else {
                if(va[i*m+j] !=(va[j*m+i] -
                1)){
                    antisimetrica = 0;
                    break;
                }
            }
        }
    }
    if(antisimetrica){
        return (3);
    }
    return (0);
   } else {
        return (-1);
   }
}
```

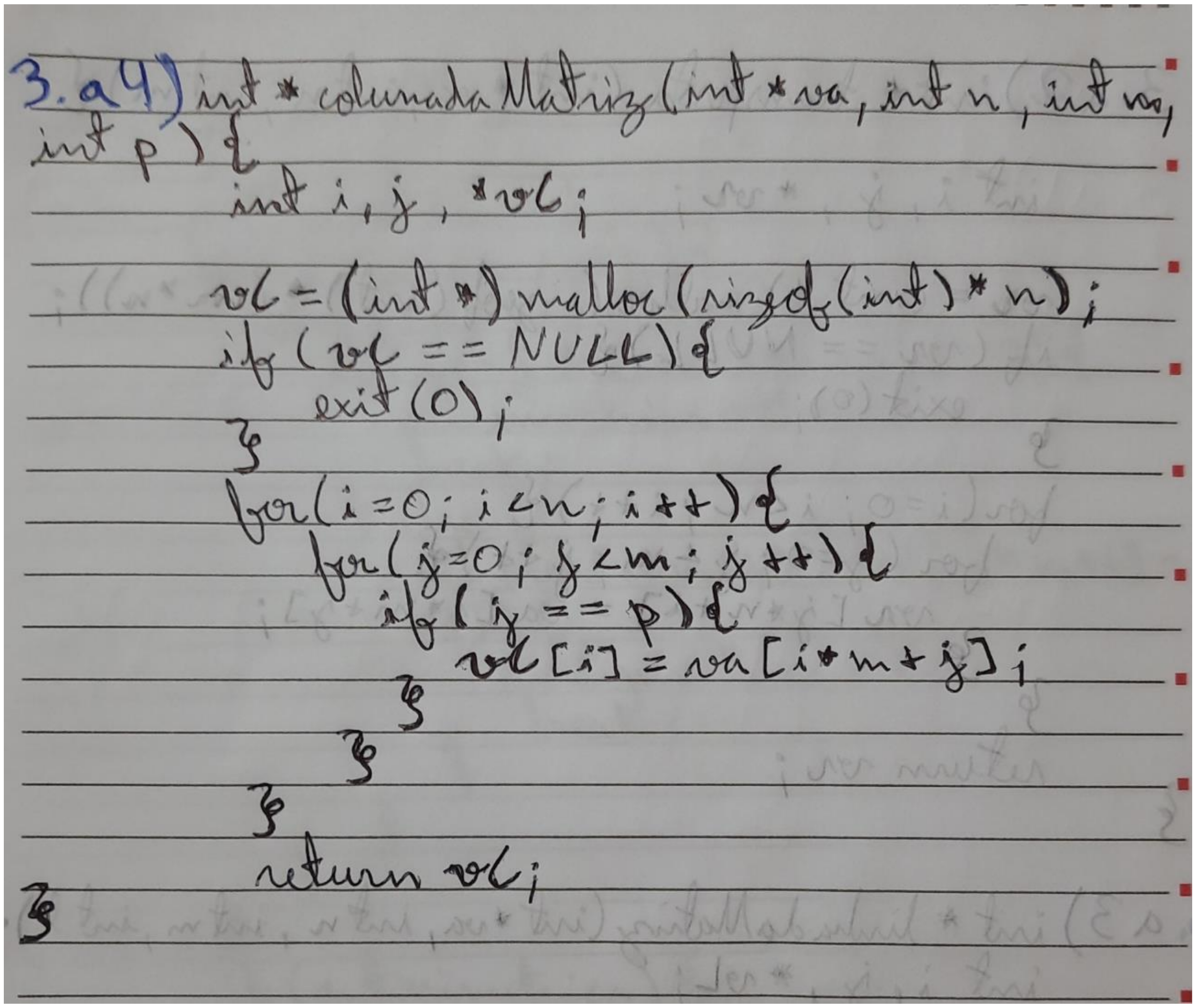
```
3.a2) int* transporta (int *va, int n, int m){

      int i, j, *vr;

      vr = (int *) malloc (sizeof (int) * (m*n));
      if (vr == NULL){
          exit(0);
      }
      for(i=0; i<n; i++){
          for (j=0; j<m; j++){
              vr[j*n+i] = va[i*m+j];
          }
      }
      return vr;
}

3.a3) int * linhadaMatriz (int *va, int n, int m, int l){
      int i, j, *vL;

      vL = (int *) malloc (sizeof(int) * m);
      if (vL == NULL){
          exit(0);
      }
      for (i=0; i<n; i++){
          for(j=0; j<m; j++){
              if (i==l){
                  vL[j] = va[i*m+j];
              }
          }
      }
}     return vL;
```



3.a4)
```
int * colunadaMatriz(int *va, int n, int m,
int p) {
        int i, j, *vc;

        vc = (int *) malloc(sizeof(int) * n);
        if (vc == NULL) {
            exit(0);
        }
        for(i=0; i<n; i++) {
            for(j=0; j<m; j++) {
                if(j == p) {
                    vc[i] = va[i*m+j];
                }
            }
        }
        return vc;
}
```

3.a5) int * diagonaldaMatriz (int *va, int n, int m){

```
    int i, j, tam, *vD;

    if(n<m){
        tam = n;
    } else {
        tam = m;
    }

    vD = (int *) malloc(sizeof(int) * tam);
    if(vD == NULL){
        exit(0);
    }
    for(i=0; i<n; i++){
        for (j=0; j<m; j++){
            if(i == j){
                vD[i] = va[i*m+j];
            }
        }
    }

    return vD;
}
```