Introducción a la Programación y Computación 1 Sección E

Ing. MSc. Neftalí Calderón

```
int binSearch(int vector[], dato) {
    int inf = 0;
    int sup = vector.lenght()-1;
    int centro = (inf+sup)/2;

    while ((vector[centro] != dato) && (inf <= sup)) {
        if (dato > vector[centro]) {
            inf = centro+1;
        }
        else {
            sup = centro-1;
        }
        centro = (inf+sup)/2;
    }

    if (vector[centro] == dato) {
        return(centro);
    }
    else {
        return(-1000);
    }
}
```

```
int binSearch(int vector[], dato, inf, sup) {
    centro = (inf+sup)/2;

    if (vector[centro] == dato) {
        return(centro);
    }
    else if (inf > sup) {
        return(-1000);
    }
    else if (dato > vector[centro]) {
        buscar(vector[], dato, centro+1, sup);
    }
    else {
        buscar(vector[], dato, inf, centro-1);
    }
}
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
95	80	70	93	60	10	5	25	82	15	14	22	21	75	72	65
80	95	70	93	60	10	5	25	82	15	14	22	21	75	72	65
80	70	95	93	60	10	5	25	82	15	14	22	21	75	72	65
80	70	93	95	60	10	5	25	82	15	14	22	21	75	72	65
80	70	93	60	95	10	5	25	82	15	14	22	21	75	72	65
80	70	93	60	10	95	5	25	82	15	14	22	21	75	72	65
80	70	93	60	10	5	95	25	82	15	14	22	21	75	72	65
80	70	93	60	10	5	25	95	82	15	14	22	21	75	72	65
80	70	93	60	10	5	25	85	95	15	14	22	21	75	72	65
80	70	93	60	10	5	25	85	15	95	14	22	21	75	72	65
80	70	93	60	10	5	25	85	15	14	95	22	21	75	72	65
80	70	93	60	10	5	25	85	15	14	22	95	21	75	72	65
80	70	93	60	10	5	25	85	15	14	22	21	95	75	72	65
80	70	93	60	10	5	25	85	15	14	22	21	75	95	72	65
80	70	93	60	10	5	25	85	15	14	22	21	75	72	95	65
80	70	93	60	10	5	25	85	15	14	22	21	75	72	65	95
70	80	93	60	10	5	25	85	15	14	22	21	75	72	65	95
70	80	60	93	10	5	25	85	15	14	22	21	75	72	65	95
70	80	60	10	93	5	25	85	15	14	22	21	75	72	65	95
70	80	60	10	5	93	25	85	15	14	22	21	75	72	65	95
70	80	60	10	5	25	93	85	15	14	22	21	75	72	65	95
70	80	60	10	5	25	85	93	15	14	22	21	75	72	65	95
70	80	60	10	5	25	85	15	93	14	22	21	75	72	65	95
70	80	60	10	5	25	85	15	14	93	22	21	75	72	65	95
70	80	60	10	5	25	85	15	14	22	93	21	75	72	65	95
70	80	60	10	5	25	85	15	14	22	21	93	75	72	65	95
70	80	60	10	5	25	85	15	14	22	21	75	93	72	65	95
70	80	60	10	5	25	85	15	14	22	21	75	72	93	65	95
70	80	60	10	5	25	85	15	14	22	21	75	72	65	93	95

```
void ordenar(int vector[]) {
    int temp;

    for (int i=1; i<vector.lenght(); i++) {
        for (int j=0; j<vector.lenght()-i; j++) {
            if (vector[j] > vector[j+1]) {
                temp = vector[j];
                vector[j] = vector[j+1];
                vector[j+1] = temp;
            }
        }
    }
}
```

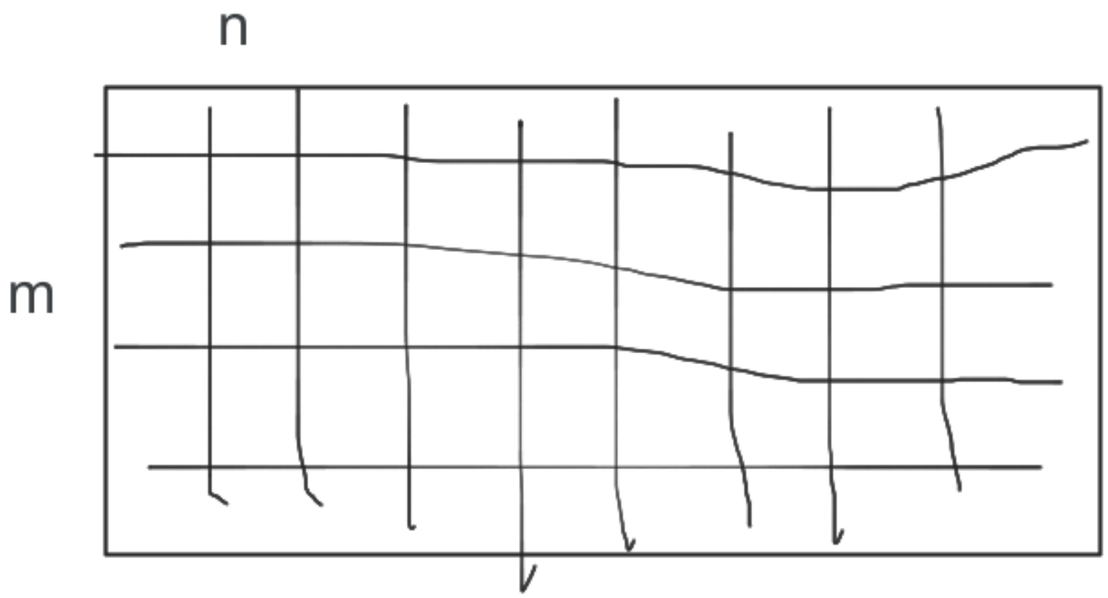
i=1
j=0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

```
void ordenar(int vector[]) {
    int temp;

    for (int i=1; i<vector.lenght(); i++) {
       for (int j=0; j<vector.lenght()-i; j++) {
           if (vector[j] > vector[j+1]) {
               temp = vector[j];
               vector[j] = vector[j+1];
               vector[j+1] = temp;
           }
       }
    }
}
```

```
void ordenar(int vector[], i) {
    int temp;

    if (i > 1) {
       for (int j=0; j<i-1; j++) {
           if (vector[j] > vector[j+1]) {
               temp = vector[j];
               vector[j] = vector[j+1];
               vector[j+1] = temp;
           }
       }
      ordenar(vector[], i-1);
    }
}
```


n
m

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
95	80	70	93	60	10	5	25	82	15	14	22	21	75	72	65
5	80	70	93	60	10	95	25	82	15	14	22	21	75	72	65
5	10	70	93	60	80	95	25	82	15	14	22	21	75	72	65
5	10	14	93	60	80	95	25	82	15	70	22	21	75	72	65
5	10	14	15	60	80	95	25	82	93	70	22	21	75	72	65