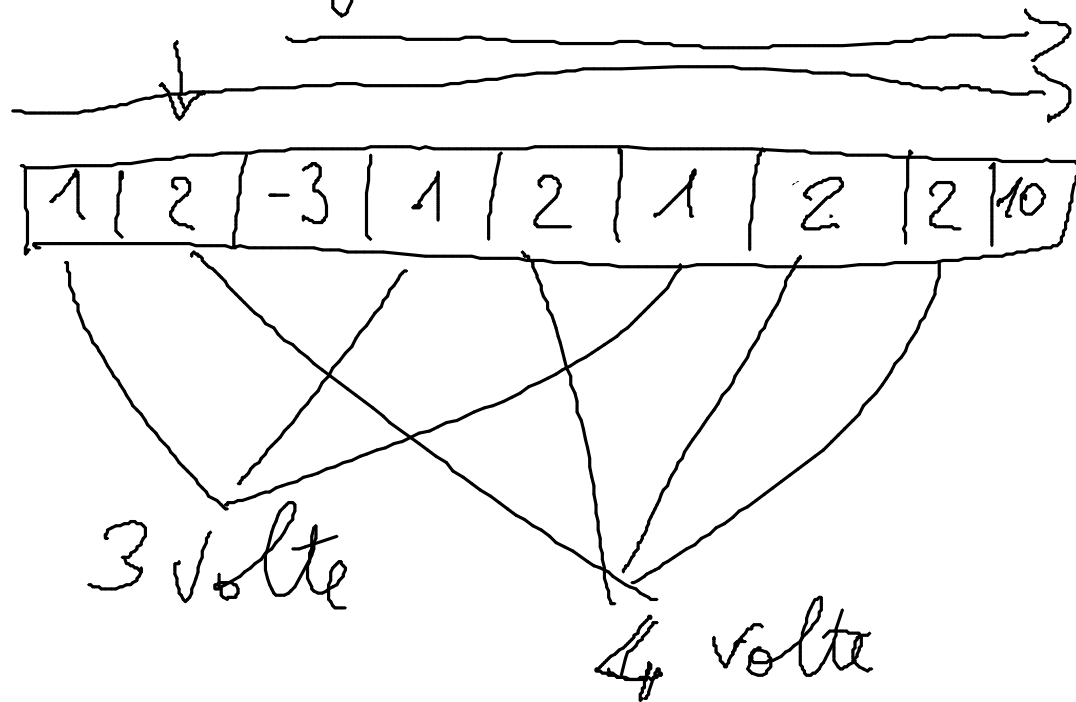


int max_occ (int a[], int dim)

che restituisce uno dei valori che occorrono in
a il maggior numero di volte



```
int max_occ (int a [], int dim)
```

```
{ int i ; int max_m, max_v;
```

```
max_v = a[0]; max_m = 1;
```

```
for (i = 0; i < dim - 1; i++)
```

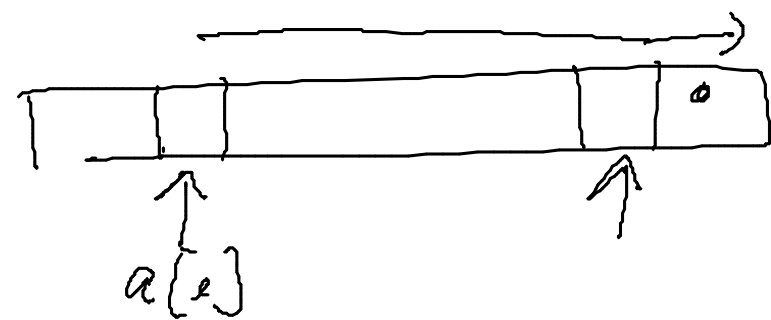
```
{ int occ = 1; int j;
```

```
for (j = i + 1; j < dim; j++)
```

```
if (a[i] == a[j]) occ = occ + 1;
```

```
if (occ > max_m) { max_m = occ; max_v = a[i]; }
```

```
return max_v;
```



Scrivere una proc. C che dato un array a di dimensione dim
 abbia tutti gli elementi $a[i]$ per i quali vale
 la seguente proprietà

$$\# \left\{ \overline{j} \in [0, n) \mid a[\overline{j}] = 0 \right\} = 2$$

1	2	0	3	4	0	0	0	0	1	-3	10
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑

Dato i

l'insieme degli indici \overline{j} compresi tra 0 e i esclusi tali
 che $a[\overline{j}] = 0$

```
void arena (int a[], int dum)
```

```
{ int i=0; int n0=0;
  while (i < dum && n0 < 2)
  { if (a[i] == 0) n0 = n0 + 1;
    i = i + 1;
  }
```

```
  while (i < dum && n0 < 3)
  { if (a[i] == 0) n0 = n0 + 1;
    else a[i] = 0;
    i = i + 1;
  }
```

```
}
```

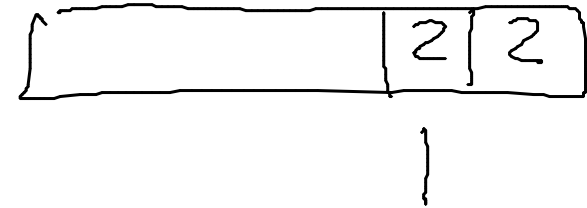
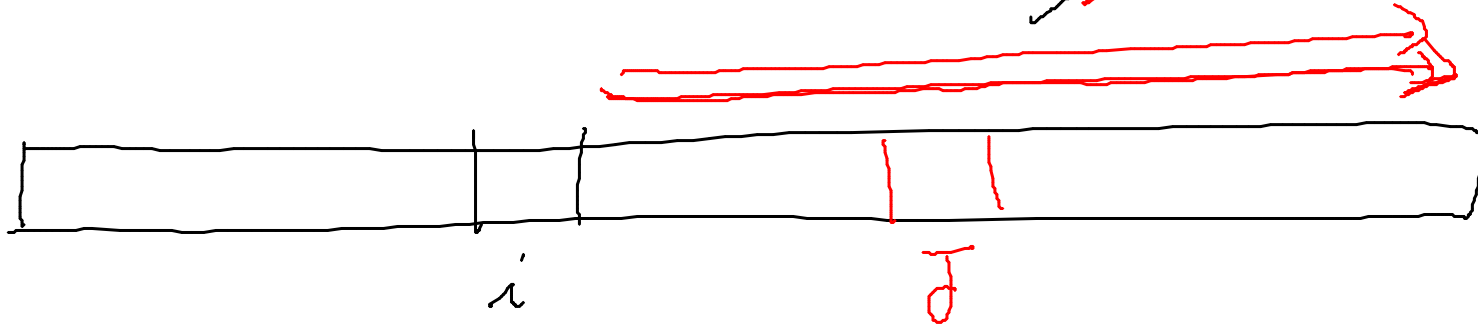
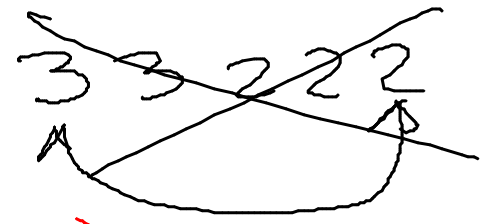
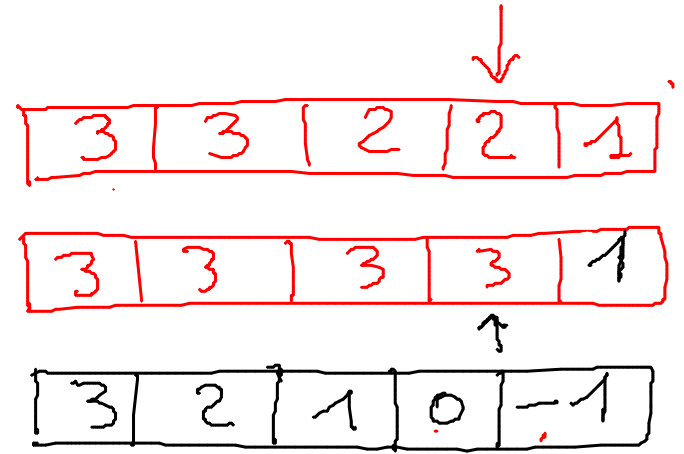
```
if (a[i] == 0) c = c + 1;
else c
if (c == 2) a[i] = 0;
```

```
for (i=0, i < dum && n < 3; i++)
```

a dim restituisce 1 se e solo se
 è verificata la proprietà

$(\forall i \in [0, \text{dim}-1]).$

$((\exists j \in [i+1, \text{dim}). a[i] > a[j]) \wedge$
 $(\forall k \in [i+1, \text{dim}). a[i] \geq a[k]))$



```

int check ( int a [], int dem)
{
    int i = 0; int perogni = 1;
    while ( i < dem - 1 && perogni )
    {
        int j = i + 1; int minore = 0; int minuz = 1;
        while ( j < dem && !minore ) if ( a[j] < a[i] ) minore = 1;
                                     else j = j + 1;
        j = i + 1;
        while ( j < dem && minuz ) if ( a[j] <= a[i] ) j = j + 1;
                                     else minuz = 0;

        perogni = minore && minuz;
        i = i + 1;
    }
    return perogni;
}

```

```
int check (int a [], int dim)
```

```
{ int i=0; int noncrescente = 1;
```

```
  while (i < dim-1 && noncrescente)
```

```
    if (a[i] >= a[i+1]) i = i+1;
```

```
    else noncrescente = 0;
```

```
  return noncrescente && a[dim-2] > a[dim-1];
```

```
}
```