

Array a di dimensione dim

$\exists j \in [0, dim)$.

$$\left(\# \{ i \mid i \in [0, dim) \wedge a[j] = a[i] \} = 3 \right)$$

```
int conte (int el, int a[], int dim)
{
    int c = 0;
    for (int i = 0; i < dim; i++)
        if (a[i] == el) c = c + 1;
    return c;
}
```

```
int check (int a[], int dim)
{
    int trovato = 0;
    int i = 0;
    while (i < dim && !trovato)
        if (conte(a[i], a, dim) == 3) trovato = 1;
        else i++;
    return trovato;
}
```

```
↳ { int c = 0;
    for (int j = 0; j < dim; j++)
```

```
if (a[j] == a[i]) c = c + 1;
if (c == 3) trovato = 1;
} else i++;
```

a di dimensione dim

$$\exists j \in (\emptyset, \text{dim}-1).$$

$$\left(\# \left\{ k \mid k \in [\emptyset, j) \wedge a[k] = a[j] \right\} = 1 \right)$$

$$\left(\# \left\{ k \mid k \in (j, \text{dim}) \wedge a[k] = a[j] \right\} = 1 \right)$$

```

int conta ( int el, int a[], int inizio, int fine )
           [inizio, fine)
{
    int c = 0;
    for ( int i = inizio; i < fine; i++ )
        if ( a[i] == el ) c = c + 1;
    return c;
}
    
```

```

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

```

{
    int trovato = 0;
    int i = 1;
    while ( i < dim - 1 && ! trovato )
        if ( conta ( a[i], a, 0, i) == 1 &&
             conta ( a[i], a, i+1, dim) == 1 ) trovato = 1;
        else i++;
    return trovato;
}
    
```

and conditional

```
return trovato;
```

Se è non form condizionale

```
if (cont(a[i], a, 0, i) == 1)
```

```
    if (cont(a[i], a, i+1, dim) == 1)
```

```
        trovato = 1;
```

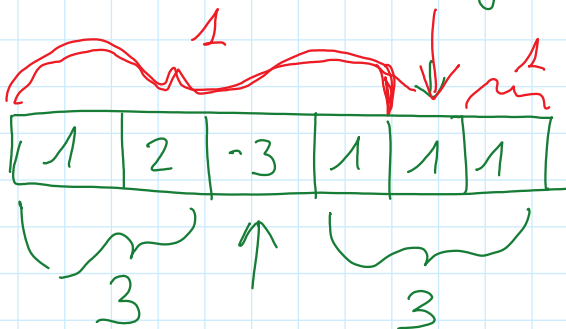
```
    else i++;
```

```
else i++;
```

a di dimensione dim

giovedì 5 ottobre 2017 16:07

$$\exists j \in (\emptyset, \text{dim}-1).$$
$$\left(\sum_{i=0}^{j-1} a[i] = \sum_{i=j+1}^{\text{dim}-1} a[i] \right)$$



```
int somma (int a[], int inizio, int fine)
           (inizio, fine)
```

```
{ int s = 0;
  for (int i = inizio; i < fine; i++)
    s = s + a[i];
  return s;
}
```

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

```
{ int trovato = 0;
  int i = 1;
  while (i < dim-1 && ! trovato)
    if (somma(a, 0, i) == somma(a, i+1, dim))
```

```
        ^      trovato = 1;  
    else    i++;  
    return trovato;  
}
```

a di dimensione dim

$\exists j \in [1, \text{dim}-1]$ $(\emptyset, \text{dim}-1)$

$(\forall i \in [\emptyset, j). a[j] < a[i]) \quad \checkmark$

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

3	3	3	3
---	---	---	---

10	11	4	5
----	----	---	---

↑

Controllare che un valore sia minore maggiori di tutti quelli dell'array a con indice in $(\text{inizio}, \text{fine})$

```

int maggiori (int el, int a[], int inizio, int fine)
{
    int ok = 1;
    int i = inizio;
    while (i < fine && ok)
        if (el <= a[i]) ok = 0;
        else i++;
    return ok;
}

```

minore

>=

```

int check (int a[], int dim)
{
    int trovato = 0;
}

```

```

int trovato = 0;
int i = 1;
while (i <= dim-1 && !trovato)
    if (minore(a[i], a, 0, i)
        || ← or in C (condizionale)
        maggiore(a[i], a, i+1, dim)) trovato = 1;
    else i++;
return trovato;

```

or } *condizionale* cor

$A \text{ cor } B \equiv \text{if } A \text{ then true}$
 $\text{else } B$

Controllare che due array, a e b , di dimensioni dim_a e dim_b , rispettivamente hanno gli stessi elementi (anche con molteplicità diverse)

a

1	2	3
---	---	---

b

3	1	2	3	5
---	---	---	---	---

$$(\forall i \in [0, dim_a). \exists j \in [0, dim_b). a[i] = b[j])$$

^

$$(\forall j \in [0, dim_b). \exists i \in [0, dim_a). b[j] = a[i])$$

int member (int el, int a[], int dim)

```
{ trovato = 0;
  int i = 0;
```

```
  while (i < dim && !trovato)
```

```
    if (a[i] == el) trovato = 1;
```

```
    else i++;
```

```
  return trovato;
```

```
}
```

```
int check (int a[], int dima,  
           int b[], int dimb)
```

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

```
  while (i < dima && ok)
```

```
    if (!member (a[i], b, dimb)) ok = 0;
```

```
    else i++;
```

```
  if (ok)
```

```
    { int j = 0;
```

```
      while (j < dimb && ok)
```

```
        if (!member (b[j], a, dima)) ok = 0;
```

```
        else j++;
```

```
      return ok;
```

```
    }
```

```
  else return ok;
```

```
}
```

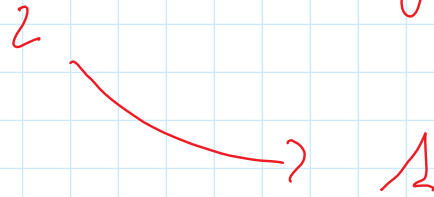
~~0~~

Controllare che a e b abbiano gli stessi elementi con la stessa molteplicità

a [3 | 2 | 2 | 3]

b [2 | 3 | 3 | 3]

⇒ ~~✓~~
false



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

int conta (int el, int a[], int dim)
{ int c = 0;

for (int i = 0; i < dim; i++)
if (a[i] == el) c = c + 1;

return c;
}

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

{ int ok = 1;
int i = 0;

```
while (i < dim && ok)
  if (cont(a[i], a, dim) ==
      cont(a[i], b, dim)) i++;
  else ok = 0;
```

```
return ok;
```

```
}
```

$$\exists j \in [1, d_m - 1].$$

$$\left(\sum_{i=0}^{j-1} a(i) = a(j) \right) \wedge$$

$$\left(\sum_{i=j+1}^{d_m-1} a(i) = a(j) \right)$$