

array a con indici nell'intervallo $[1, \text{dim})$
formula che ci dice se l'array è ordinato
in modo crescente è la seguente

$$\left(\forall i \in [0, \text{dim}-1) . a[i] < a[i+1] \right)$$

per ogni elemento deve valere la
proprietà

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

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

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

```
    if (a[i] >= a[i+1]) tutti_ok = 0;
```

```
    else i = i+1;
```

```
  } return tutti_ok;
```

→ ok

tutti_ok == 1

≡ i++;

→ {0, 1} ≡ {false, true}

funzione che controlli se un valore compare o meno in un array

member el a dim
 supponiamo di dover cercare un elemento el in un array a di dimensione dim

$$(\exists i \in [0, \text{dim}). a[i] = \text{el})$$

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

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

not → ! trovato

```
main ()
```

```
{
    int element;
    int a [10];
    scanf ("%d" & element);
}
```

```
scanf ("%d", & elements);  
read (a, 10); ← legge tutti gli element  
dell' array
```

```
printf ("%d", member (elements, a, 10));
```

```
if (member (elements, a, 10))
```

```
    printf ("d'elemento %d compare in a", element);  
else    "    "    "    non compare in a", element);
```

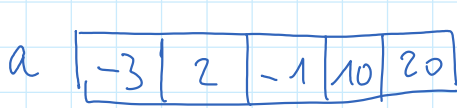
```
int member (int el, int a[], int dim)
{
    int i = 0;
    int trovato = 0;
    while (i < dim && !trovato)
        ;
}
```

L'importanza del nome delle variabili:
(che deve essere significativo per il lettore)

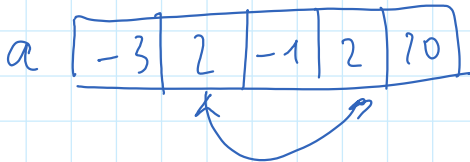
Numero Di Element
Somma Degli Element

```
int topolino (int mippo, int minnie,  
             int qui)
;
;
```

Dato un array controllare se tutti gli elementi siano distinti



→ true → 1



→ false → ∅

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

$\neg (\exists j \in [i+1, \text{dim}) . a[j] = a[i])$

```

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

Dove cercare un elemento = el nell'array

Dato cercare un elemento $= el$ nell'array
 a , nell'intervallo degli indici: $(inizio, fine)$

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

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

```
  int ok = 1;
```

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

```
    if (member(a[i], a, i + 1, dim)) ok = 0;
```

```
    else i++;
```

```
  return ok;
```

```
}
```

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

stiamo utilizzando il for per una
iterazione indeterminata

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

$$\left(\forall i \in [0, \text{dim} - 1], \right. \\ \left. \left(\forall j \in [i+1, \text{dim}). a[i] \neq a[j] \right) \right)$$

$$\neg \left(\exists x. P \right) \equiv \left(\forall x. \neg P \right)$$

$(\forall i \in [\bar{i}, \text{dim}-1]. \neg (\exists j \in [i+1, \text{dim}). a[i] = a[j]))$

int distinct (int a[], int dim)

{ int i = 0;

int ok = 1;

while (i < dim - 1 && ok)

{ int j = i + 1;

int trovato = 0;

while (j < dim && !trovato)

if (a[j] == a[i]) trovato = 1;

else j++;

if (trovato) ok = 0;

else i++;

} return ok;

}

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

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

int diversi (int el, int a[], int inizio, int fine)

{ int i = inizio;

int ok = 1;

while (i < fine && ok)

if (a[i] == el) ok = 0;
else i++;

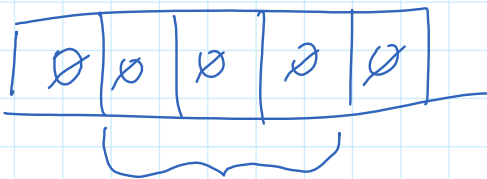
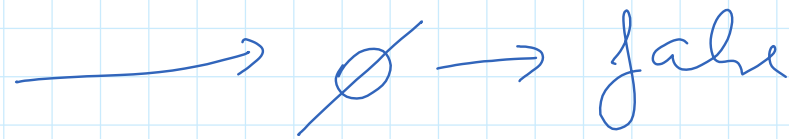
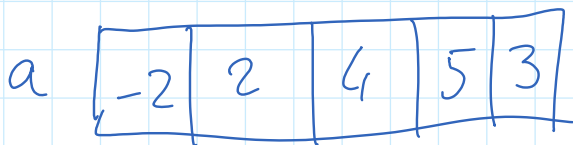
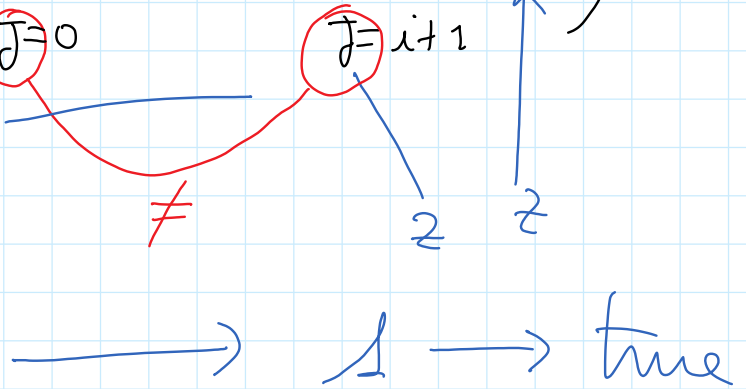
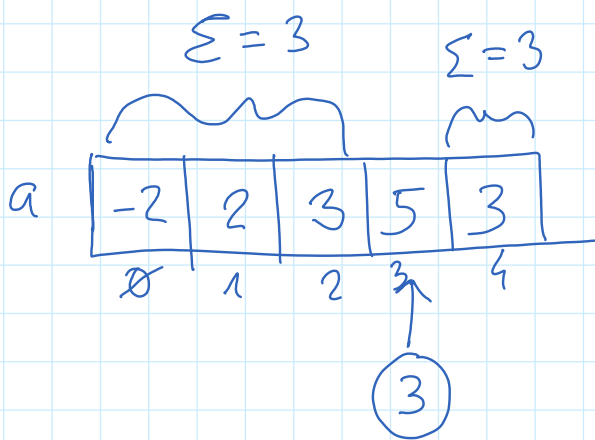
} return ok;

↳ if (a[i] != el) i++;
else ok = 0;

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

Date un array a , di dimensione dim ,
 controllo le verità delle formule seguenti

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



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

deve restituire la somma degli elementi di a
con indici nell'intervallo [inizio, fine)

```

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

```

Dati due array, a e b , di dimensioni dim_a e dim_b , rispettivamente vogliamo scrivere una funzione che restituisca il valore di verità delle seguente formule:

$(\exists i \in [0, dim_a).$

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

```

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



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

```
{  
  int i = 0;  
  int trovato = 0;  
  while (i < dima && ! trovato)  
    if (member(a[i], b, dimb)) trovato = 1;  
    else i++;  
  return trovato;  
}
```