

Si scriva una funzione  $C$  che, dati due array,  $a$  e  $b$ , entrambi di dimensione  $dim$ , restituisce il valore di verità delle formule

$$\left( \forall i \in [1, dim), \right. \\ \left. b[i] = \# \{ j \mid j \in [0, i) \wedge a[j] \geq 0 \} \right)$$

a

1	-3	2	-4	1
---	----	---	----	---

b

10	1	1	2	2
0	1	2	3	4

int countmeg (int a[], int fine)

{

il numero degli elem<sup>t</sup> di  $a$  con valore  
 nell'intervallo  $[0, fine)$  che sono  
 maggiori o uguali a 0.

}

```

int contermag (int a [], int fine)
{
    int i;
    int c = 0;
    for (i = 0; i < fine; i++)
        if (a[i] >= 0) c++;
    return c;
}

```

```

int formule (int a [], int b [], int dim)
{
    int i = 1;
    int ok = 1;
    while (i < dim && ok)
        if (b[i] == contermag (a, i)) i++;
        else ok = 0;
    return ok;
}

```

dato un array  $a$ , di dimensione  $dim_a$ , e un array  $b$ , di dimensione  $dim_b$ , restituisce il valore di verità delle formule:

$$\underline{(\forall i \in [0, dim_a])}$$

$$(a[i] > 0 \Rightarrow$$

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

member

$$\rightarrow A \Rightarrow B \equiv \neg A \vee B$$

$$\neg(\neg A \vee B) \equiv A \wedge \neg B$$

```
int member (int el, int a[], int dim)
{
    ...
}
```

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

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

```
while (i < dima && ok)
{ if (a[i] > 0)
  { if (!member (a[i], b, dimb)) ok = 0;
  }
  i++;
}
```

$$A \Rightarrow B \equiv \neg A \vee B$$

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

$$\neg (\neg A \vee B) \equiv A \wedge \neg B$$

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

```
return ok;
```

}

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

restituisce il valore di verità delle formule:

$(\exists i \in [0, \text{dim})).$

$$\# \{k \mid k \in [0, i) \wedge a[k] < 0\}$$

$$= \# \{s \mid s \in (i, \text{dim}) \wedge a[s] < 0\}$$

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

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

Formule:

$$(\exists i \in [0, \text{dim})).$$

$$(\exists j \in (i, \text{dim})).$$

$$(\# \{ k \mid k \in [0, \text{dim}) \wedge a[i] = a[k] \})$$

$$\# \{ s \mid s \in [0, \text{dim}) \wedge a[j] = a[s] \}$$

$$\geq 2)))$$

$$A > B \geq C$$

$$A > B \quad \&\&$$

$$B \geq C$$



$$\left( \exists i. \left( \exists j. P \right) \right) \text{ esistono due indici}$$
$$\equiv$$
$$\left( \exists j. \left( \exists i. P \right) \right)$$

---

$$\left( \exists i. \left( \forall j. P \right) \right)$$
$$\neq$$
$$\left( \forall j. \left( \exists i. P \right) \right)$$

```
int conte (int el, int a, dim)
{
    int i;
    int c = 0;

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

    return c;
}
```

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

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

```
  int trovati = 0;
```

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

```
  { int j = i + 1;
```

```
    while (j < dim && ! trovati)
```

```
      if (contae (a[j], a, dim) >= 2 &&
```

```
          (contae (a[i], a, dim) >
```

```
              contae (a[j], a, dim))) trovati = 1;
```

```
      else j++;
```

```
    } i++;
```

```
  return trovati;
```

```
}
```