

```

bool[] V
int size
int dim

SET Set(int m)
[
    SET t ← new SET
    t.size ← 0
    t.dim ← m
    t.V ← [false] * m

    return t
]

SET contains(int x)
[
    if  $1 \leq x \leq \textit{dim}$  then
    |   return V[x]
    else
    |   return false
]

int size
[
    return size
]

insert(int x)
[
    if  $1 \leq x \leq \textit{dim}$  then
    |   if not V[x] then
    |   |   size++
    |   |   V[x] ← true
]

remove(int x)
[
    if  $1 \leq x \leq \textit{dim}$  then
    |   if V[x] then
    |   |   size++
    |   |   V[x] ← false
]

```

```

SET union(SET A, SET B)
[
    // crea un insieme della capacità max
    SET C ← Set(max(A.dim, A.dim))

    // inserisci gli elementi di A
    from i ← 1 until A.dim do
    |   if A.contains(i) then
    |   |   C.insert(i)

    // inserisci gli elementi di B
    from i ← 1 until B.dim do
    |   if A.contains(i) then
    |   |   C.insert(i)
]

SET intersection(SET A, SET B)
[
    // crea un insieme della capacità min
    SET C ← Set(min(A.dim, A.dim))

    from i ← 1 until min(A.dim, A.dim) do
    |   // se è contenuto in entrambi
    |   if A.contains(i) and B.contains(i) then
    |   |   C.insert(i) // aggiungilo
]

SET difference(SET A, SET B)
[
    SET C ← Set(A.dim)

    from i ← 1 until A.dim do
    |   // se è contenuto A e non in B
    |   if A.contains(i) and not B.contains(i)
    |   |   then
    |   |   |   C.insert(i) // aggiungilo
]

```