

## PSEUDOCODIFICA LISTA DINAMICA COSTITUITA DA n NODI

**ALGORITMO** ListaNodiDinamica

**TIPO NODO** = RECORD

Info : INT

pNext: PUNTATORE A NODO

**FINE RECORD**

**PROCEDURA** main( )

pTesta, pCur, PNew: PUNTATORE A NODO

i, n: INT

continua: BOOL

**INIZIO**

*/\* costruzione della lista \*/*

*/\* check del numero di nodi minimo per la lista \*/*

**RIPETI**

Leggi (n)

**FINCHE'** (n ≥ 2)

Alloca (pNew, DimensioneDi (NODO)) (1)

**SE** (pNew ≠ NULL)

**ALLORA**

Leggi (pNew→Info)

pCur ← pNew (2)

pTesta ← pCur (3)

i ← 2

continua ← VERO

**MENTRE** (continua = VERO) AND (i ≤ n) **ESEGUI**

Alloca (pNew, DimensioneDi (NODO)) (4)

**SE** (pNew ≠ NULL)

**ALLORA**

Leggi (pNew→Info)

(pCur→pNext) ← pNew (5)

pCur ← pNew (6)

i ← i + 1

**ALTRIMENTI**

continua ← FALSO

Scrivi ("Errore di allocazione altri nodi")

**FINE SE**

**FINE MENTRE**

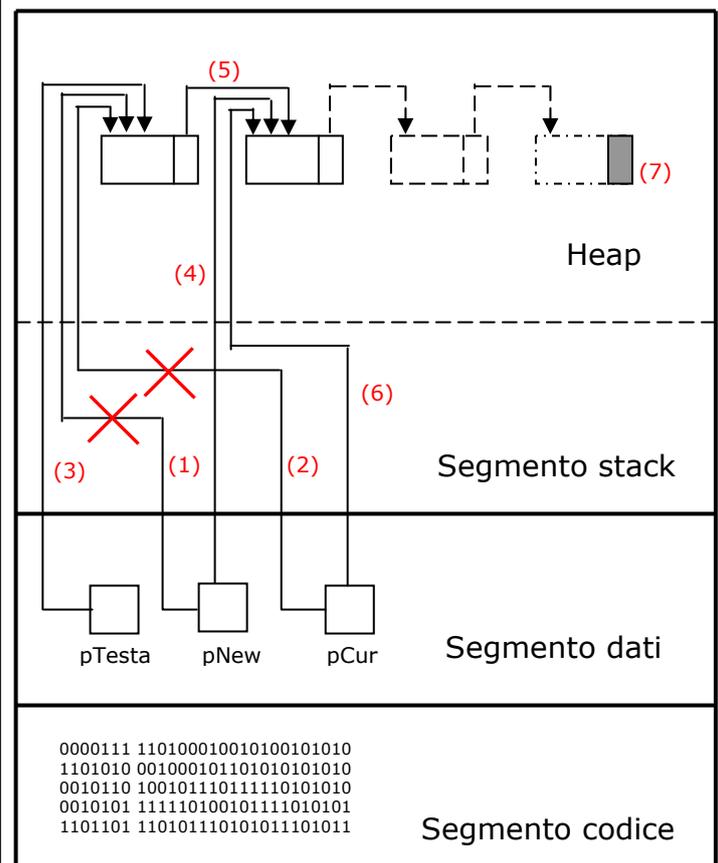
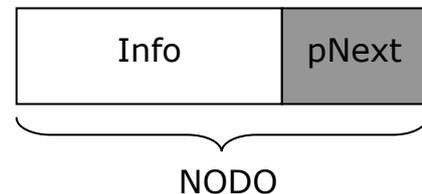
(pCur→pNext) ← NULL (7)

**ALTRIMENTI**

continua ← FALSO

Scrivi ("Errore di allocazione primo nodo")

**FINE SE**



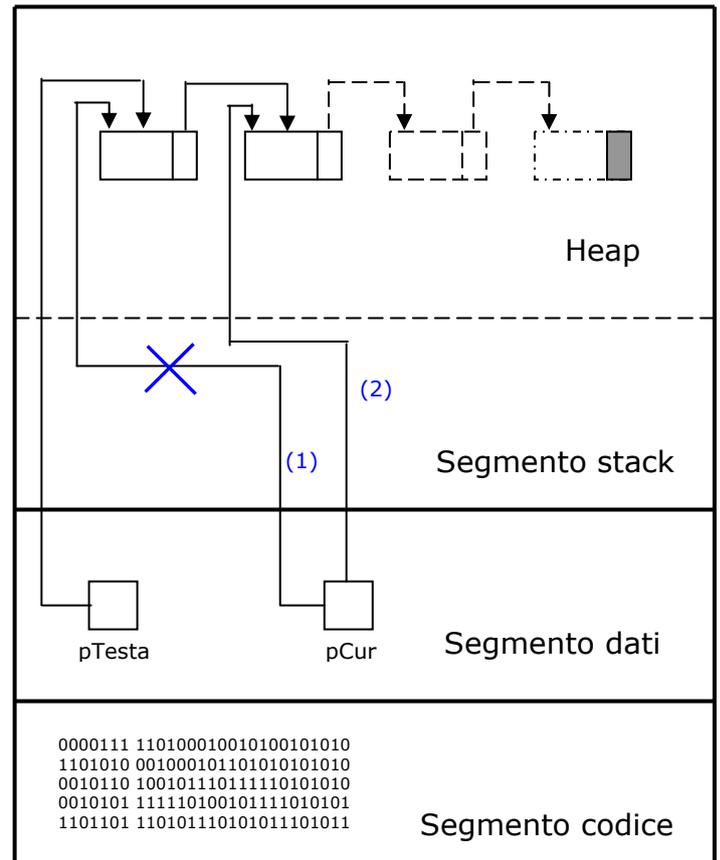
SE (continua = VERO)  
**ALLORA**

```

/* visualizzazione della lista */
pCur ← pTesta           (1)
MENTRE (pCur ≠ NULL) ESEGUI
  Scrivi (pCur → Info)
  pCur ← (pCur → pNext) (2)
FINE MENTRE
    
```

```

/* visualizzazione alternativa della lista */
pCur ← pTesta
MENTRE (pCur ≠ NULL) ESEGUI
  Scrivi ((*pCur).Info)
  pCur ← ((*pCur).pNext)
FINE MENTRE
    
```



```

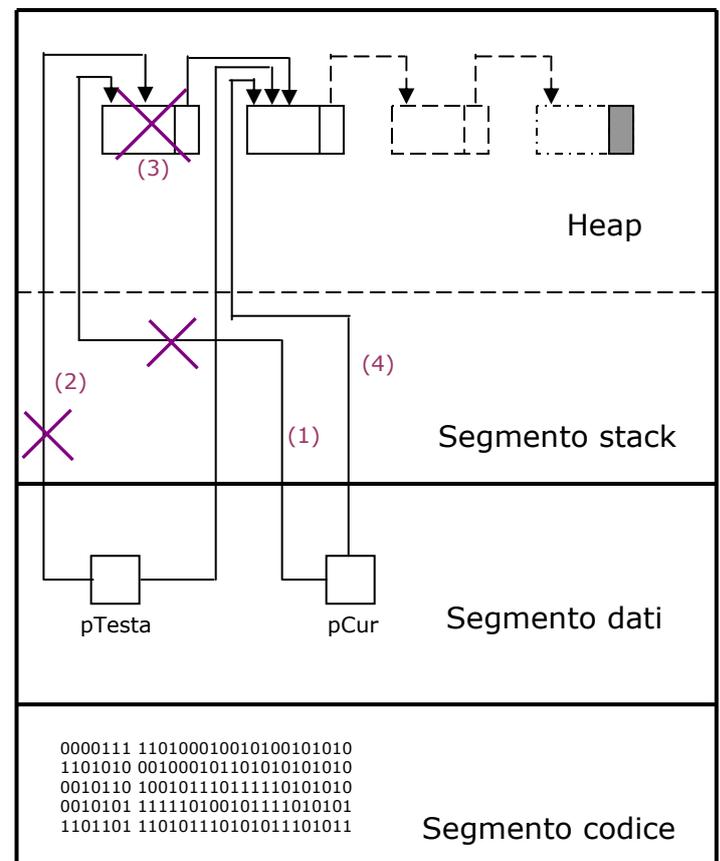
/* deallocazione della lista */
pCur ← pTesta           (1)
MENTRE (pCur ≠ NULL) ESEGUI
  pTesta ← (pCur → pNext) (2)
  Dealloca (pCur)         (3)
  pCur ← pTesta          (4)
FINE MENTRE
    
```

```

/* deallocazione alternativa della lista */
pCur ← pTesta
MENTRE (pCur ≠ NULL) ESEGUI
  pTesta ← ((*pCur).pNext)
  Dealloca (pCur)
  pCur ← pTesta
FINE MENTRE
    
```

**FINE SE**

**FINE**



## FLOW-CHART

