

```
/*  
 * Corso di Fondamenti di Informatica  
 * Esercizio:  
 * tema d'esame  
 */
```

```
/* Dato il seguente frammento di programma, determinare i valori stampati dalle  
printf: */
```

```
#include <stdio.h>
```

```
int main()  
{  
    int i;  
    int arr[10];  
    int *parr[5];  
  
    for(i=4; i>=0; i--)  
        *(parr+i)=arr+9-2*i;
```

```
    for(i=0; i<10; i++)  
        arr[i]=20+i;
```

```
    printf("\n%d", *(parr-1));
```

```
    for(i=0; i<10; i++)  
    {  
        i++;  
        arr[i]=i+arr[i];  
    }
```

```
    printf("\n%d", *((parr+3)+1));  
    printf("\n");
```

```
    return (0);  
}
```

```
/* -----  
Soluzione:
```

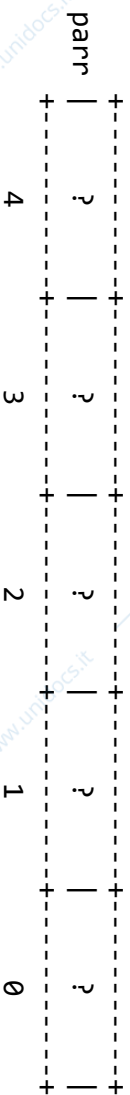
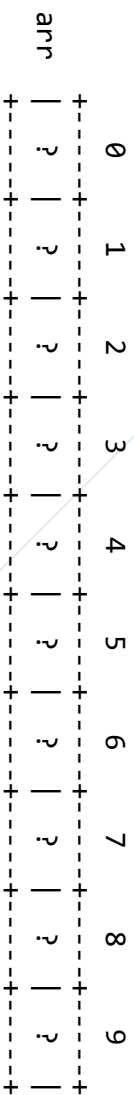
(ricordare che se Arr è un array, l'identificatore Arr corrisponde a &Arr[0] e Arr+k corrisponde -per l'aritmetica dei puntatori- a &Arr[k])

arr è un array di 10 interi.

parr è un array di 5 puntatori ad intero.

Vediamo passo per passo quale sia il contenuto dei due array durante l'esecuzione del programma.

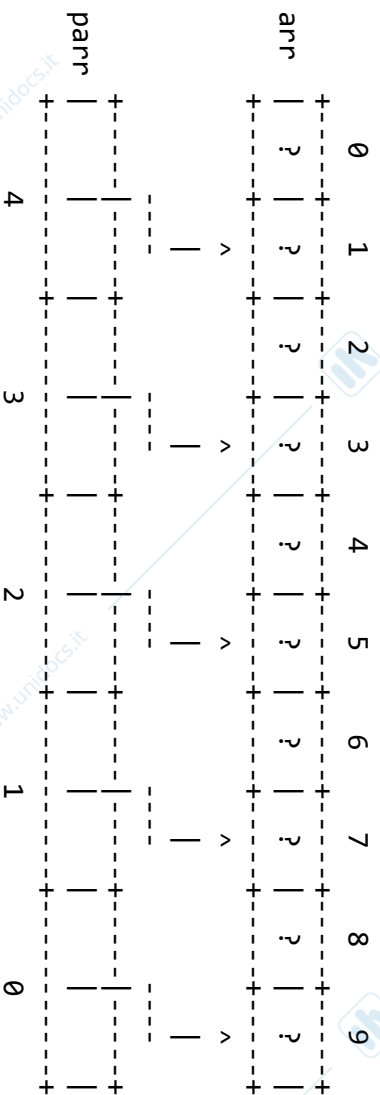
Appena dichiarati, gli array non sono inizializzati e dunque contengono dati ignoti. Rappresentando per comodità parr e arr in versi opposti, si ha dunque



Dopo l'esecuzione di

```
for(i=4; i<=9; i--)  
*(parr+i)=arr+9-2*i;
```

(che modifica gli elementi di parr) la situazione è la seguente:



Infatti
parr = &parr[0]

```
parr+i = &parr[i]
*(parr+i) = parr[i]
```

e

```
arr = &arr[0]
arr+9-2*i = &arr[0+9-2*i] = &arr[9-2*i]
```

Dopo l'esecuzione di

```
for(i=0; i<10; i++)
arr[i]=20+i;
```

il contenuto di parr non è variato, mentre quello di arr si'. Ora la situazione e' la seguente:

```
0 1 2 3 4 5 6 7 8 9
arr | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
+---+---+---+---+---+---+---+---+---+---+
      ^         ^         ^         ^         ^
      |         |         |         |         |
      +---+---+---+---+---+---+---+---+---+
parr |   |   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+---+
      4         3         2         1         0
```

A questo punto viene fatta la prima stampa. Per sapere cosa viene stampato a schermo è possibile "espandere" l'espressione `*(parr-1)`, tenendo conto del significato dei vari simboli (incluso l'identificativo dell'array parr):

```
*(parr-1) = (*(parr[0]-1) = *(parr[0]-1) = *(parr[9]-1) = *(parr[8]) =
= arr[8] = 28
```

Dunque viene stampato a schermo il numero 28.
Dopo l'esecuzione di

```
for(i=0; i<10; i++)
{
    i++;
    arr[i]=i+arr[i];
}
```

il contenuto di parr non è variato, mentre quello di arr si'. Ora la situazione è' la seguente:

	0	1	2	3	4	5	6	7	8	9
arr	20	22	22	26	24	30	26	34	28	38
		^		^		^		^		^
parr										
		-----		-----		-----		-----		-----
	4		3		2		1		0	

Quindi la successiva istruzione printf stampa a schermo

```

*(*(parr+3)+1) = (*( &parr[0]+3)+1) = (*( &parr[3])+1) = *(parr[3]+1) =
= *( &arr[3]+1) = *( &arr[4]) = arr[4] = 24
----- */

```