

```
/*
 * Corso di Fondamenti di Informatica
 * Esercizio:
 * tema d'esame - esercizio di analisi del codice
 */

/* Indicare tutto ciò che il seguente programma C stampa a schermo, nell'ordine in cui tali stampe avvengono */
```

```
#define L 10
#include <stdio.h>

int *PrimaFunzione(int In[]);
int *SecondaFunzione(int *In);

void main()
{
    int E[L], N;
    for (N=0; N<L; ++N) E[N]=L/2-N+1;
    printf("%d", *SecondaFunzione(PrimaFunzione(E)));
}
```

```
int *PrimaFunzione(int In[])
{
    printf("%d", *In);
    return(In+2);
}
```

```
int *SecondaFunzione(int *In)
{
    printf("%d", *(In-1));
    return(In);
}
```

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

Dopo l'esecuzione del ciclo for, l'array E contiene i valori
6 5 4 3 2 1 0 -1 -2 -3

L'istruzione

```
printf("%d", *SecondaFunzione(PrimaFunzione(E)));
```

esegue chiamate a 3 diverse funzioni.

In ordine di chiamata, tali funzioni sono:

1. PrimaFunzione, alla quale viene passato il parametro attuale E, che per definizione di array vale $\&(E[0])$;
2. SecondaFunzione, alla quale viene passato il valore appena restituito da PrimaFunzione, che (considerando il codice di PrimaFunzione) vale $\&(E[0]) + 2 = \&(E[2])$;
3. printf, alla quale viene passato il dato puntato dal valore appena restituito da SecondaFunzione, che (considerando il codice di SecondaFunzione) vale $\&(E[2])$.

Considerando il codice di PrimaFunzione e SecondaFunzione, si ottiene che con questi parametri attuali:

1. PrimaFunzione stampa $\&(E[0])$ ovvero $E[0]$, e dunque il valore 6;
2. SecondaFunzione stampa $\&(E[2]) - 1$, ovvero $\&(E[1])$, ovvero $E[1]$, e dunque il valore 5;
3. printf stampa $\&(E[2])$, ovvero $E[2]$, e dunque il valore 4.

Il programma stampa quindi

```
654
```

```
*/
```