

Fondamenti di Informatica T-1

Esercitazione 1

Grammatiche

Complemento a 2

Analisi

Record di attivazione

Contenuto di questa esercitazione

- Esercizi sulle grammatiche
- Esercizi sulla rappresentazione dei numeri
- Esercizi di analisi
- Esercizi sui record di attivazione
- Esercizi di sintesi

Esercizio - Grammatiche (1)

Espressioni algebriche

$G = \langle VT, VN, P, S \rangle$, dove:

$VT = \{ +, -, *, /, (,), 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 \}$

$VN = \{ \langle E \rangle, \langle T \rangle, \langle F \rangle, \langle \text{num} \rangle, \langle \text{cifra} \rangle, \langle \text{cifra-non-nulla} \rangle \}$

$S = \langle E \rangle$

Esercizio - Grammatiche (1)

Espressioni Algebriche

P = {

<E> ::= <E> + <T> | <E> - <T> | <T>

<T> ::= <T> * <F> | <T> / <F> | <F>

<F> ::= <cifra> | (<E>)

<cifra> ::= 0 | <cifra-non-nulla>

<cifra-non-nulla> ::= 1|2|3|4|5|6|7|8|9

}

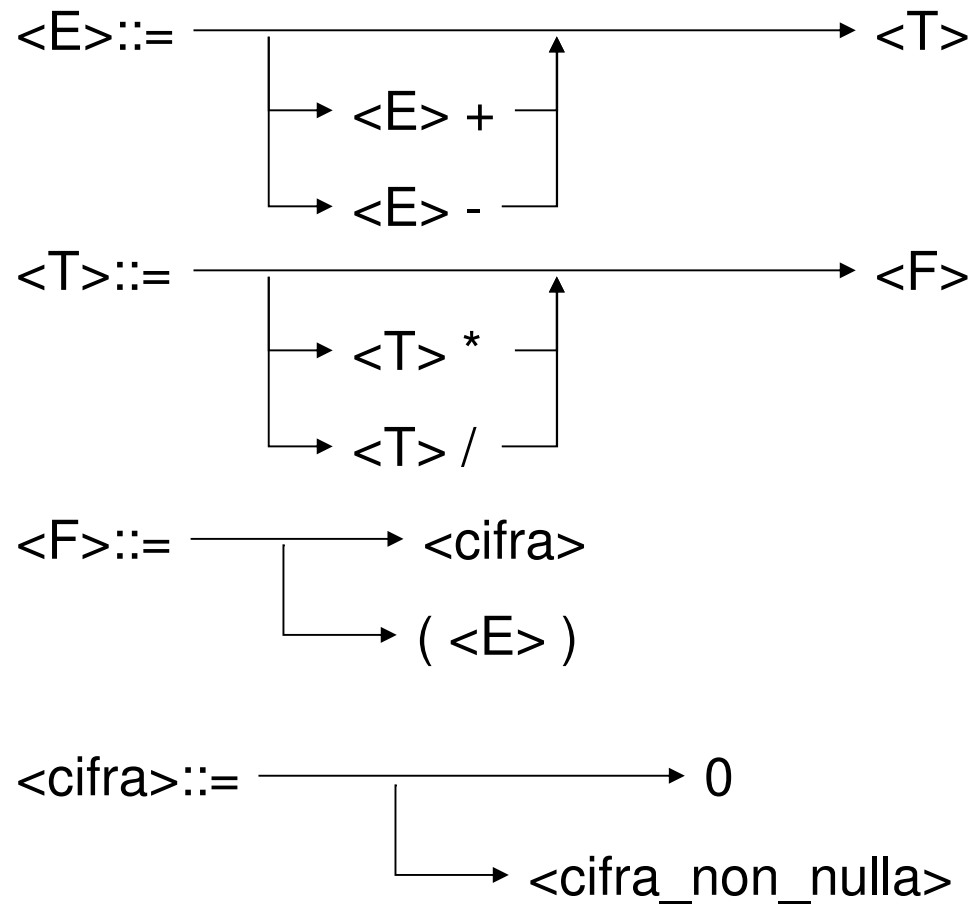
Disegnare il diagramma sintattico di tale grammatica. Determinare poi se le seguenti frasi fanno parte del linguaggio generato da questa grammatica o no, e disegnarne l'albero di derivazione sintattica, e la derivazione left-most:

- 1. 1 + 4 / 2**
- 2. (1 + 4) / 2**

Esercizio - Grammatiche (1)

Soluzione

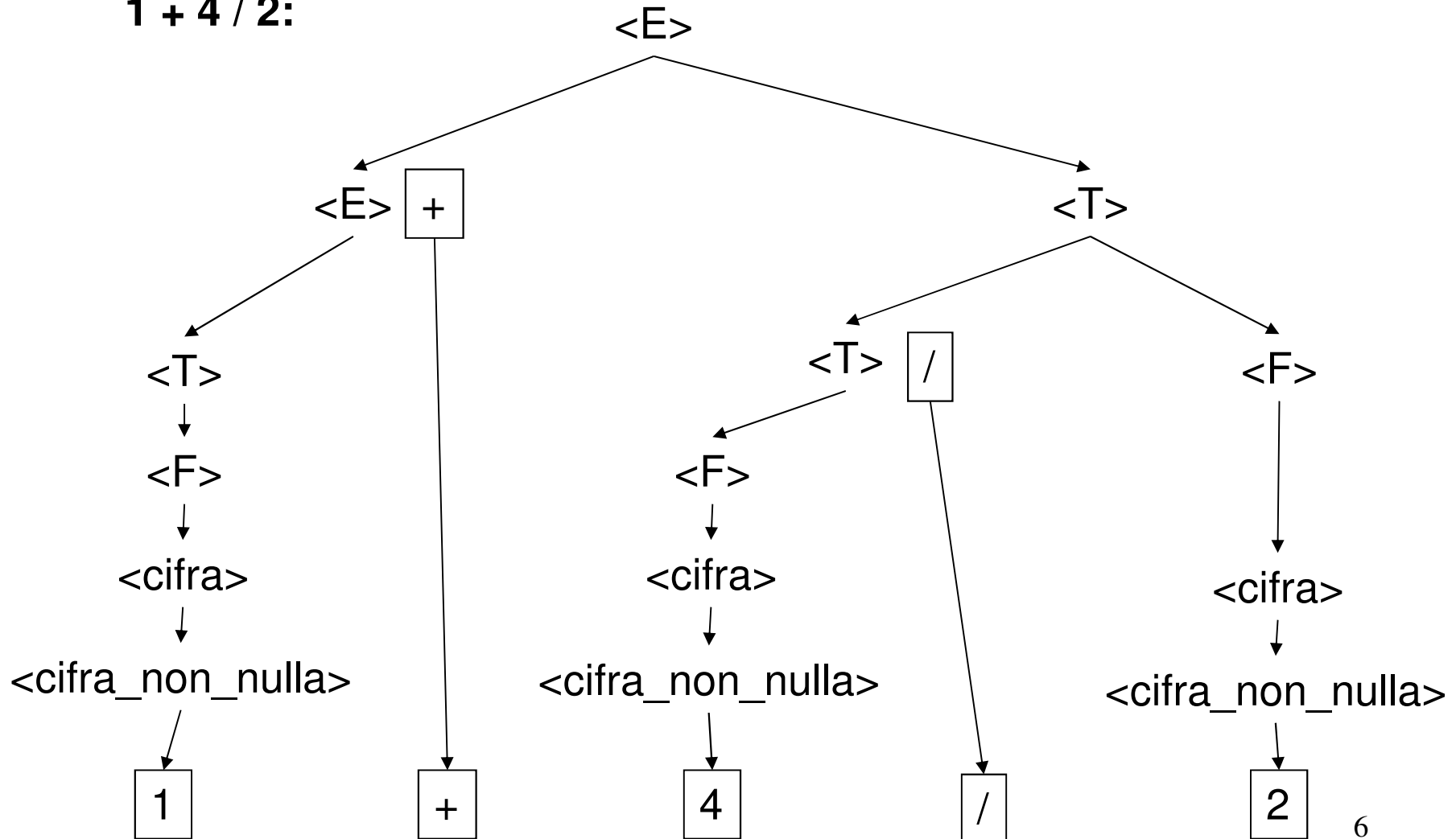
Diagramma sintattico:



Esercizio - Grammatiche (1)

Soluzione

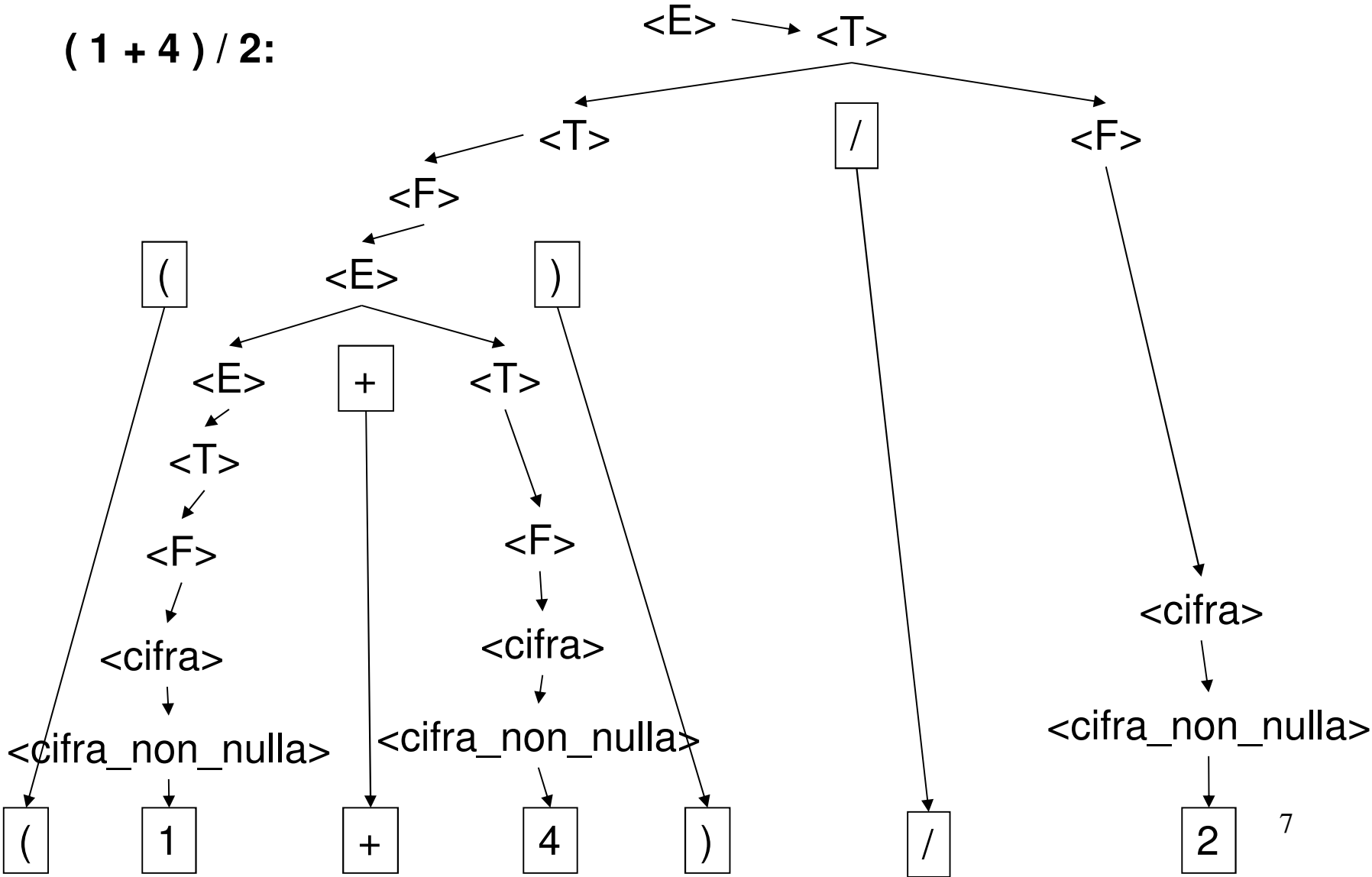
1 + 4 / 2:



Esercizio - Grammatiche (1)

Soluzione

(1 + 4) / 2:



Esercizio - Grammatiche (1)

Soluzione

Derivazione left-most:

1. 1 + 4 / 2

$\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle$	$\rightarrow \langle T \rangle + \langle T \rangle$
$\rightarrow \langle F \rangle + \langle T \rangle$	$\rightarrow \langle cifra \rangle + \langle T \rangle$
$\rightarrow \langle cifra_non_nulla \rangle + \langle T \rangle$	$\rightarrow 1 + \langle T \rangle$
$\rightarrow 1 + \langle T \rangle / \langle F \rangle$	$\rightarrow 1 + \langle F \rangle / \langle F \rangle$
$\rightarrow 1 + \langle cifra \rangle / \langle F \rangle$	$\rightarrow 1 + \langle cifra_non_nulla \rangle / \langle F \rangle$
$\rightarrow 1 + 4 / \langle F \rangle$	$\rightarrow 1 + 4 / \langle cifra \rangle$
$\rightarrow 1 + 4 / \langle cifra_non_nulla \rangle$	$\rightarrow 1 + 4 / 2$

Esercizio - Grammatiche (1)

Soluzione

Derivazione left-most:

2. $(1 + 4) / 2$

$\langle E \rangle \rightarrow \langle T \rangle \rightarrow \langle T \rangle / \langle F \rangle$
 $\rightarrow \langle F \rangle / \langle F \rangle \rightarrow (\langle E \rangle) / \langle F \rangle$
 $\rightarrow (\langle E \rangle + \langle T \rangle) / \langle F \rangle \rightarrow (\langle T \rangle + \langle T \rangle) / \langle F \rangle$
 $\rightarrow (\langle F \rangle + \langle T \rangle) / \langle F \rangle \rightarrow (\langle cifra \rangle + \langle T \rangle) / \langle F \rangle$
 $\rightarrow (\langle cifra_non_nulla \rangle + \langle T \rangle) / \langle F \rangle \rightarrow (1 + \langle T \rangle) / \langle F \rangle$
 $\rightarrow (1 + \langle F \rangle) / \langle F \rangle \rightarrow (1 + \langle cifra \rangle) / \langle F \rangle$
 $\rightarrow (1 + \langle cifra_non_nulla \rangle) / \langle F \rangle \rightarrow (1 + 4) / \langle F \rangle$
 $\rightarrow (1 + 4) / \langle cifra \rangle \rightarrow (1 + 4) / \langle cifra_non_nulla \rangle$
 $\rightarrow (1 + 4) / 2$

Esercizio - Grammatiche (2)

Compito del 13 Settembre 2007

La stringa data inizia col simbolo terminale '1' e termina col simbolo terminale 'b'.

Per poter ottenere una tale stringa, dovrebbe esserci una produzione con F come primo simbolo non terminale e G come ultimo simbolo non terminale...

... ma nessuna delle produzioni date ha tali caratteristiche. Infatti l'unica produzione che può terminare con G è $S ::= GXY$, ma G come primo simbolo non terminale è incompatibile con la stringa data.

La stringa data non appartiene quindi al linguaggio generato dalla grammatica data.

Esercizio - RAPPRESENTAZIONE (3)

Un elaboratore rappresenta i numeri interi su 8 bit in **complemento a 2**. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

$$39 + (-91)$$

Esercizio - RAPPRESENTAZIONE (3)

+39 -> 00100111

91 -> 01011011

10100100

10100101 -> -91

Conversione del valore assoluto
in base 2

Inversione dei bit

Aggiungo 1

Si esegue la somma:

00100111 +

10100101

11001100

11001100

00110011

00110100 -> 52 (base 10)

Complemento a 2 – quando c'è overflow?

Nella rappresentazione a complemento a 2, l'overflow può accadere solo in conseguenza di:

- Somma di due numeri positivi
- Somma di due numeri negativi

*L'overflow si verifica quando uno dei due ultimi bit di riporto è a uno, ma non lo sono entrambi (**xor degli ultimi due bit di riporto**)*

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri positivi senza overflow (caso su 4 bit):

(riporto) 0 0 0 0 `xor(0, 0) = 0, non c'è overflow!`

3 -> 0011

4 -> 0100

0111

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri positivi con overflow (caso su 4 bit):

(riporto) 0 1 1 1
3 -> 0011
5 -> 0101

1000

$\text{xor}(0, 1) = 1$, OVERFLOW!!!

Sommando due numeri positivi, è stato modificato il bit del segno...
... deve essere successo qualcosa!!!

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri negativi senza overflow (caso su 4 bit):

(riporto) 1 1 0 0
-3 -> 1 1 0 1
-4 -> 1 1 0 0

$\text{xor}(1, 1) = 0$, no overflow

1 0 0 1

Il bit del segno è rimasto invariato...
... ed entrambi i riporti sono ad 1...

Complemento a 2 – quando c'è overflow?

Consideriamo il caso di due numeri negativi con overflow (caso su 4 bit):

(riporto)		1 0 0 0	xor(1, 0) = 1, OVERFLOW!!!
-3	->	1101	
-6	->	1010	

		0111	Il bit del segno è stato modificato... ...cosa è successo ?

Esercizio - RAPPRESENTAZIONE (4)

Un elaboratore rappresenta i numeri interi su 8 bit in **complemento a 2**. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

$$(-13) - (-91)$$

Esercizio - RAPPRESENTAZIONE (4)

13 -> 00001101
11110010
11110011 -> -13

-(-91) -> 01011011

(riporti) -> 1 1 1 1 1 1 1 1
-13 -> 1 1 1 1 0 0 1 1 +
-91 -> 0 1 0 1 1 0 1 1

0 1 0 0 1 1 1 0

01001110 -> 78 in base 10

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;
    → v = 0;
      { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4
i
v = 0
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

N = ~~4~~5
i
v = 0

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

N = ~~4~~5
i
v = ~~0~~5

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

N = ~~4~~54
i
v = ~~0~~5

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

N = ~~4~~54
i
v = ~~0~~51

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

N = ~~4543~~
i
v = ~~051~~

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    → for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4543
i = 0
v = 051
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

N = ~~4543~~
i = 0
v = ~~0510~~

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    → for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4543
i = 01
v = 0510-1
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    → for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4543
i = 012
v = 0510-1-2
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    → for (i=0; i<N; i++)
        v = v - 1;
    for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4543
i = 0123
v = 0510-1-2
```

Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    → for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4543
i = 01230
v = 0510-1-2
```


Esercizio analisi (5)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori di v e di N al termine dell'esecuzione? (si motivi opportunamente la risposta data)

```
int main(void)
{
    int N=4, i, v;

    v = 0;
    { ++N; }

    do {
        if ((N%2) == 1)
            v = v + N;
        else
            v = v - N;

        N--;
    } while (v > 4);

    for (i=0; i<N; i++)
        v = v - 1;
    → for (i=0; i < (i?N:0); )
        v = 10;
    return (0);
}
```

```
N = 4543
i = 01230
v = 0510-1-2
```

```
if (i != 0)
    N
else
    0
```

Esercizio analisi (5)

Il programma compila correttamente, e al termine dell'esecuzione le variabili v ed N valgono rispettivamente -2 e 3

- Le prime due istruzioni pongono v a 0, e poi incrementano il valore di N , che da 4 viene a valere 5.
- Il primo ciclo che poi si incontra nel codice è di tipo `do..while`, perciò almeno una volta viene eseguito. In particolare, l'istruzione di `if` controlla: se N è dispari allora v viene incrementato di N , altrimenti v viene decrementato di N . Il ciclo termina quando la condizione ($v > 4$) non vale più. Alla prima iterazione dunque v viene a valere 5, ed N è decrementato a 4; alla seconda iterazione v viene a valere 1 ed N decrementato a 3: Siccome la condizione di continuazione del ciclo non è più verificata, il ciclo si interrompe.
- Il secondo ciclo che si incontra decrementa di 1 il valore di v , per N volte. Quindi al termine v viene a valere -2.
- Il terzo ciclo presente nel programma non viene eseguito, poichè la condizione è immediatamente non verificata.

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }
```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    → { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

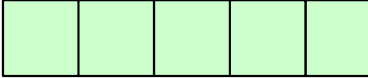
```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

N = ~~4~~5

V = 

i

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }
    → for (i=0; i<N; i++)
        v[i]=2*i+1;

```

N = ~~4~~5

V =

1				
---	--	--	--	--

i = 0

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }
    → for (i=0; i<N; i++)
        v[i]=2*i+1;

```

N = ~~4~~5

V =

1	3			
---	---	--	--	--

i = ~~0~~1

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }
    → for (i=0; i<N; i++)
        v[i]=2*i+1;

```

N = ~~4~~5

V =

1	3	5		
---	---	---	--	--

i = ~~0~~~~1~~2

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }
    → for (i=0; i<N; i++)
        v[i]=2*i+1;

```

N = ~~4~~5

V =

1	3	5	7	
---	---	---	---	--

i = ~~0~~~~1~~~~2~~3

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```


Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }
    → for (i=0; i<N; i++)
        v[i]=2*i+1;

```

N = ~~4~~5

V =

1	3	5	7	9
---	---	---	---	---

i = ~~0~~~~1~~~~2~~~~3~~4

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

N =	4 5				
v =	1	3	5	7	9
i =	0	1	2	3	4 50

→ for (i=0; i<N-1; i++)
 if (v[i]) F(v, i);

```
printf("Adesso N vale: %d\n", N);
for (i=1; i++ < (i?N:0); )
    printf("%d\n", v[i-1]);
return 0; }
```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;  N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

Chiamo F poichè $v[0] \neq 0$

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

Questa N non è la stessa del main!

N =	4 5					
v =	1	3	5	7	9	
i =	0	1	2	3	4 5	0

```
    for (i=0; i<N-1; i++)
        → if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;  N--;
    → *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
pos = 0
N   = 43

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
N = 45
```

4

```
v = 

|   |              |   |   |   |
|---|--------------|---|---|---|
| 1 | <del>3</del> | 5 | 7 | 9 |
|---|--------------|---|---|---|


```

```
i = 0123450
```

```
for (i=0; i<N-1; i++)
    if (v[i]) F(v, i);

```

```
printf("Adesso N vale: %d\n", N);
for (i=1; i++ < (i?N:0); )
    printf("%d\n", v[i-1]);
return 0; }
```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    → *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
pos = 1
N   = 43

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
N = 45
```

```
4 9
```

```
v = 

|   |              |              |   |   |
|---|--------------|--------------|---|---|
| 1 | <del>3</del> | <del>5</del> | 7 | 9 |
|---|--------------|--------------|---|---|


```

```
i = 01234501
```

```
for (i=0; i<N-1; i++)
    if (v[i]) F(v, i);

```

```
printf("Adesso N vale: %d\n", N);
for (i=1; i++ < (i?N:0); )
    printf("%d\n", v[i-1]);
return 0; }
```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;  N--;
    → *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
pos = 2
N   = 43

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
N = 45
      4 9 16
v = [ 1 | 3 | 5 | 7 | 9 ]
i = 0 1 2 3 4 5 0 1 2

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    → *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
pos = 3
N   = 43

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
N = 45
      4 9 16 25
v = 

|   |              |              |              |              |
|---|--------------|--------------|--------------|--------------|
| 1 | <del>3</del> | <del>5</del> | <del>7</del> | <del>9</del> |
|---|--------------|--------------|--------------|--------------|


i = 0123450123

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

Adesso N vale: 5

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

N = ~~4~~5
 4 9 16 25
v =

1	3	5	7	9
---	--------------	--------------	--------------	--------------

i = ~~0~~~~1~~~~2~~~~3~~~~4~~~~5~~~~0~~~~1~~~~2~~~~3~~4

```
→ printf("Adesso N vale: %d\n", N);
   for (i=1; i++ < (i?N:0); )
       printf("%d\n", v[i-1]);
   return 0; }
```


Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }
```

N = ~~4~~5

4 9 16 25

V =

1	3	5	7	9
---	--------------	--------------	--------------	--------------

i = ~~0~~~~1~~~~2~~~~3~~~~4~~~~5~~~~0~~~~1~~~~2~~~~3~~~~4~~1

1. if(i!=0) N else 0
2. i < N
3. i = i+1

Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Adesso N vale: 5
4

N = ~~4~~5

4 9 16 25

v =

1	3	5	7	9
---	--------------	--------------	--------------	--------------

i = ~~0~~~~1~~~~2~~~~3~~~~4~~~~5~~~~0~~~~1~~~~2~~~~3~~~~4~~2

1. if(i!=0) N else 0
2. i < N (1<5)
3. i = i+1 (i=2)



Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Adesso N vale: 5
4
9

N = ~~4~~5

4 9 16 25

v =

1	3	5	7	9
---	--------------	--------------	--------------	--------------

i = ~~0~~~~1~~~~2~~~~3~~~~4~~~~5~~~~0~~~~1~~~~2~~~~3~~~~4~~~~1~~~~2~~3



Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

```

```
    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Adesso N vale: 5
4
9
16

N = ~~4~~5

4 9 16 25

V =

1	3	5	7	9
---	--------------	--------------	--------------	--------------

i = ~~0~~~~1~~~~2~~~~3~~~~4~~~~5~~~~0~~~~1~~~~2~~~~3~~~~4~~~~1~~~~2~~~~3~~4



Esercizio analisi (6)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
void F(int v[], int pos) {
    int N=4;    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return; }

```

```
int main() {
    int N=4, v[5], i;
    { ++N; }

    for (i=0; i<N; i++)
        v[i]=2*i+1;

```

```
    for (i=0; i<N-1; i++)
        if (v[i]) F(v, i);

    printf("Adesso N vale: %d\n", N);
    for (i=1; i++ < (i?N:0); )
        printf("%d\n", v[i-1]);
    return 0; }

```

Adesso N vale: 5
4
9
16
25

N = ~~4~~5

4 9 16 **25**

v =

1	3	5	7	9
---	--------------	--------------	--------------	--------------

i = ~~0~~~~1~~~~2~~~~3~~~~4~~~~5~~~~0~~~~1~~~~2~~~~3~~~~4~~~~1~~~~2~~~~3~~~~4~~5

1. if(i!=0) N else 0
2. i < N (4<5)
3. i = i+1 (i=5)



Esercizio analisi (6)

Soluzione

Il programma compila correttamente e stampa:

Adesso N vale: 5

4

9

16

25

- la funzione **F** si limita a modificare il valore dell'array con indice (**pos+1**) sommandogli il valore contenuto all'indice **pos**,
- Nel **main**, il primo ciclo **for** provvede ad inizializzare l'array **v** ai valori {**1, 3, 5, 7, 9**}; il secondo ciclo invece invoca ripetutamente la funzione **F**, ed all'uscita dal ciclo il vettore viene a valere {**1, 4, 9, 16, 25**}.
- Infine viene stampato il il valore di **N** (che in seguito al preincremento iniziale passa da 4 a 5), e poi con un ultimo ciclo viene stampato il contenuto del vettore **v** (però solo le ultime 4 posizioni).

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    → char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

S =	P	a	p	e	r	o	n	e	\0
SS =									
i =	3								
N =	2								

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    → N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

S =	P	a	p	e	r	o	n	e	\0
SS =									
i =	3								
N =	2 31								

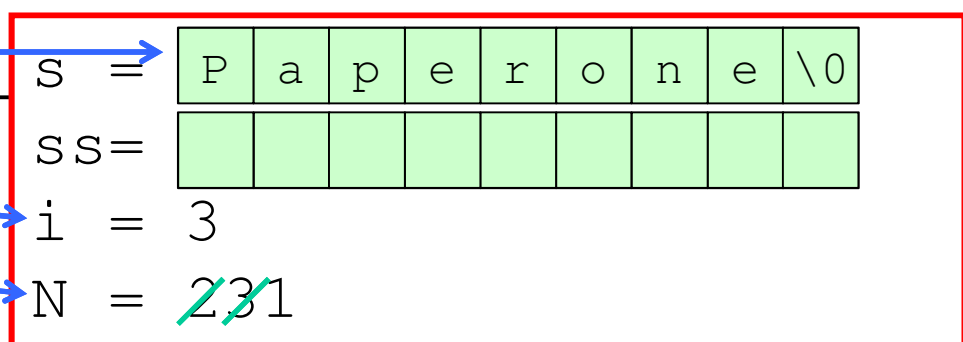
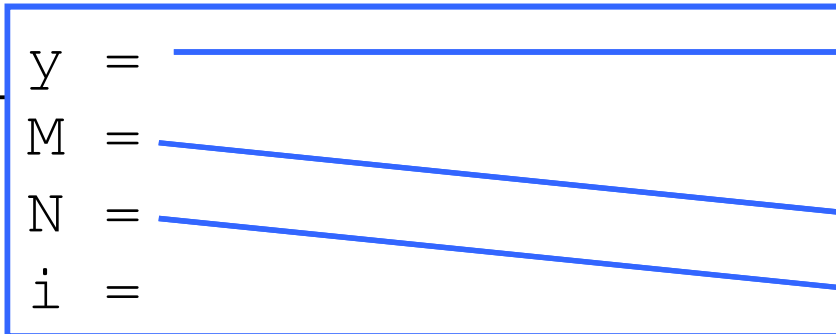
Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```



Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i>(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

Y =
M =
N =
i =

S = P a p e r o n e \0
SS =
i = 3
N = ~~2~~~~3~~2

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    → for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    → N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

Y =
M =
N =
i = 1

S = P a p e r o n e \0
SS =
i = 3
N = ~~2~~~~3~~2

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}

int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

Y =
M =
N =
i = 1

S = P a p e r o n e \0
SS =
i = 3
N = 2

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    → return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    → N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

~~Y =
M =
N =
i = 12~~

~~S = P a p e r o n e \0
SS =
i = 3
N = 2312~~

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

N vale adesso: 2

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

		e							
S =	P	a	p	e	r	o	n	e	\0
SS =									
i =	3								
N =	2 3 2								

Esercizio analisi (7)

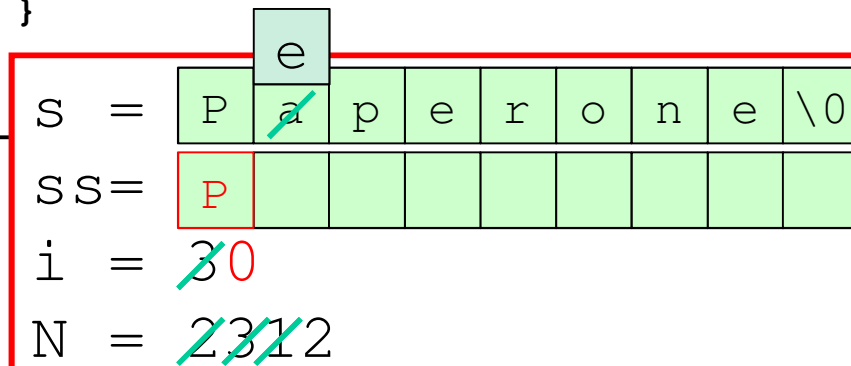
Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

N vale adesso: 2



Esercizio analisi (7)

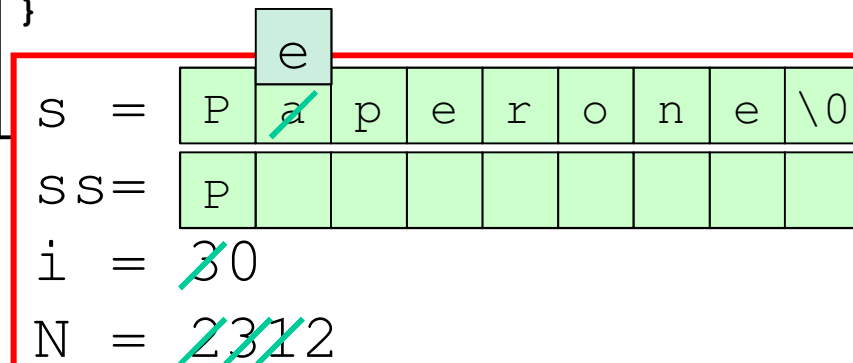
Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

N vale adesso: 2
P

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```


s = P a p e r o n e \0
ss = P
i = 3
N = 2

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

N vale adesso: 2
Pe

S =	P	a	p	e	r	o	n	e	\0
SS =	P	e							
i =	3	1							
N =	2	3	2						

Esercizio analisi (7)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 9

int Funz(char y[], int *N, int *M)
{
    int i;
    (*N)++;
    for (i=(*N)-1; i<*N; i++)
        y[i] = y[*M];
    return *N;
}
```

```
int main () {
    char s[] = "Paperone";
    char ss[DIM]; int i=3, N = 2;
    N = (++N)-2;
    N = Funz(s, &N, &i);
    printf("N vale adesso: %d\n",N);
    {
        for (i=0; i<DIM; ++i) {
            *(ss+i) = s[i];
            printf("%c", *(ss+i));
        }
    }
    return 0;
}
```

N vale adesso: 2
Peperone\0

		e									
S =	P	a	p	e	r	o	n	e	\0		
SS =	P	e	p	e	r	o	n	e	\0		
i =	3	0	1	2	3	4	5	6	7	8	9
N =	2	3	1	2							

Esercizio analisi (7)

Soluzione

Il programma compila correttamente e stampa:

N vale adesso: 2

Peperone

- La funzione **Funz** incrementa il valore riferito da **N**, e poi assegna all'elemento in posizione ***N-1** del vettore **y** il valore in posizione ***M**; restituisce infine il nuovo valore riferito da **N**
- Il **main** inizia dichiarando alcune variabili, istanziando **N** a **2** e poi modificandolo con un assegnamento al valore **1**
- Viene poi invocata la funzione **Funz**, con parametri ("**Paperone**", **1**, **3**); per quanto detto, viene modificato l'elemento in posizione **1** (**'a'**), a cui viene assegnato il valore in posizione **3** (**'e'**)
- Infine l'array **s** viene copiato (elemento per elemento) nell'array **ss**, che viene poi stampato a video

Esercizio Record di Attivazione (8)

Si consideri la seguente funzione:

```
int funzione(float a, float b) {
    a = a+b;
    b--;
    if (b > 0)
        return funzione(a, b);
    else
        return a;
}
```

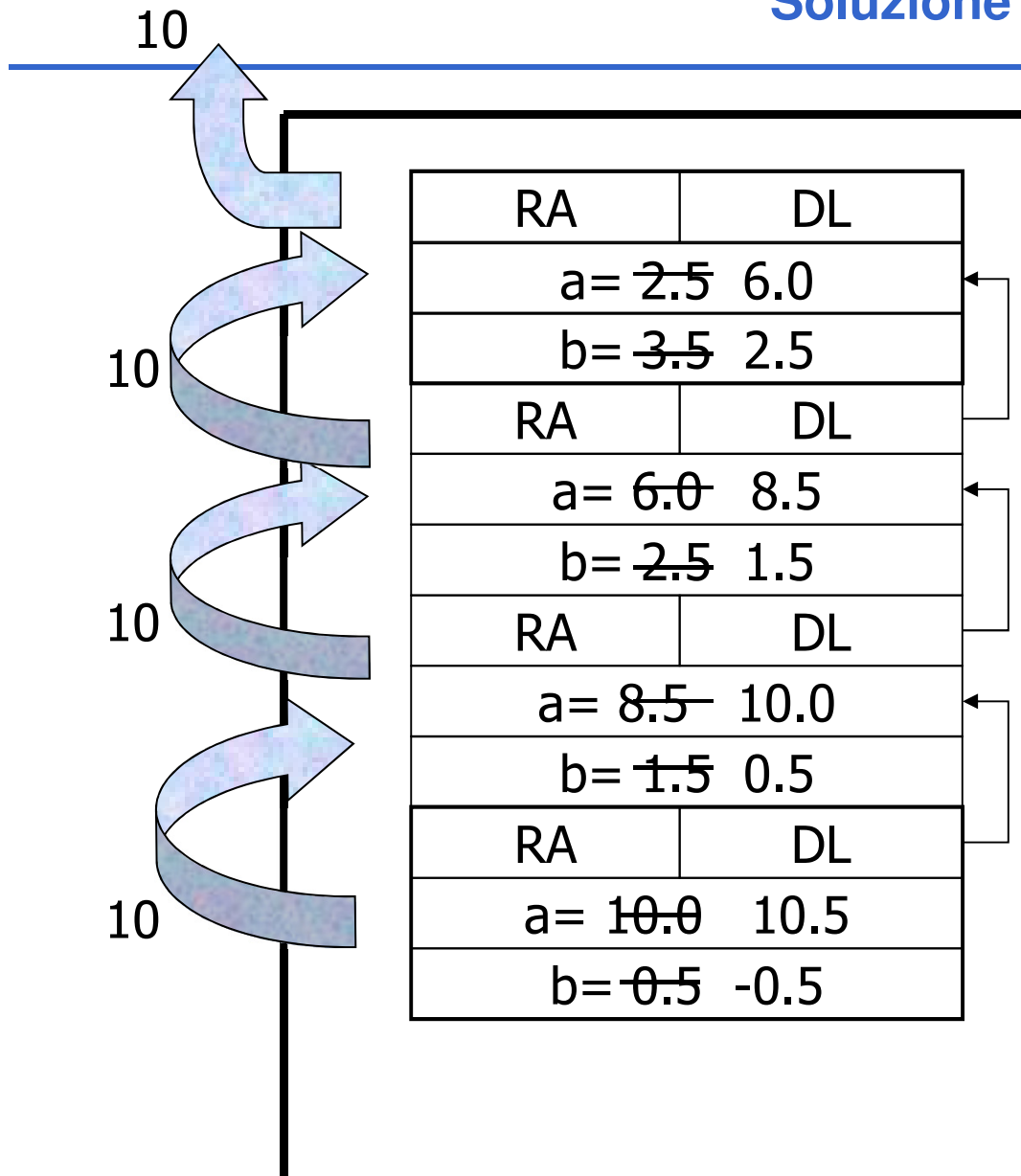
Si scriva il risultato della funzione quando invocata come:

funzione(2.5, 3.5)

e si disegnino i corrispondenti record di attivazione (si faccia particolare attenzione al fatto che la funzione restituisce un valore intero).⁶⁹

Esercizio Record di Attivazione (8)

Soluzione



```
int funzione(float a,
float b) {
    a = a+b;
    b--;
    if (b > 0)
        return
        funzione(a, b);
    else
        return a;
}
```

Invocazione
funzione(2.5, 3.5)

Esercizio Record di Attivazione (9)

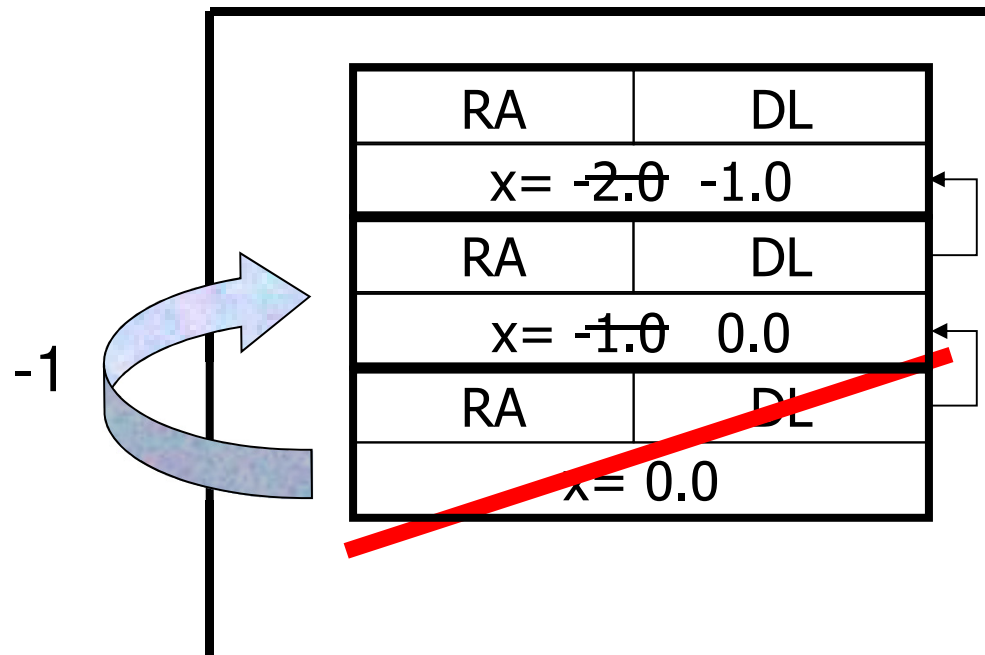
Si consideri la seguente funzione **W**:

```
double W(int x) {
    if (x<0) {
        x++;
        return W(x)+W(x/2);
        x++;
    }
    else
        return -1;
}
```

Si scriva il risultato della funzione quando invocata come **W(-2)** e si disegnino i corrispondenti record di attivazione.

Esercizio Record di Attivazione (9) Soluzione

Supponendo che la valutazione degli addendi nella somma venga fatta a partire da sinistra, si ottiene prima:



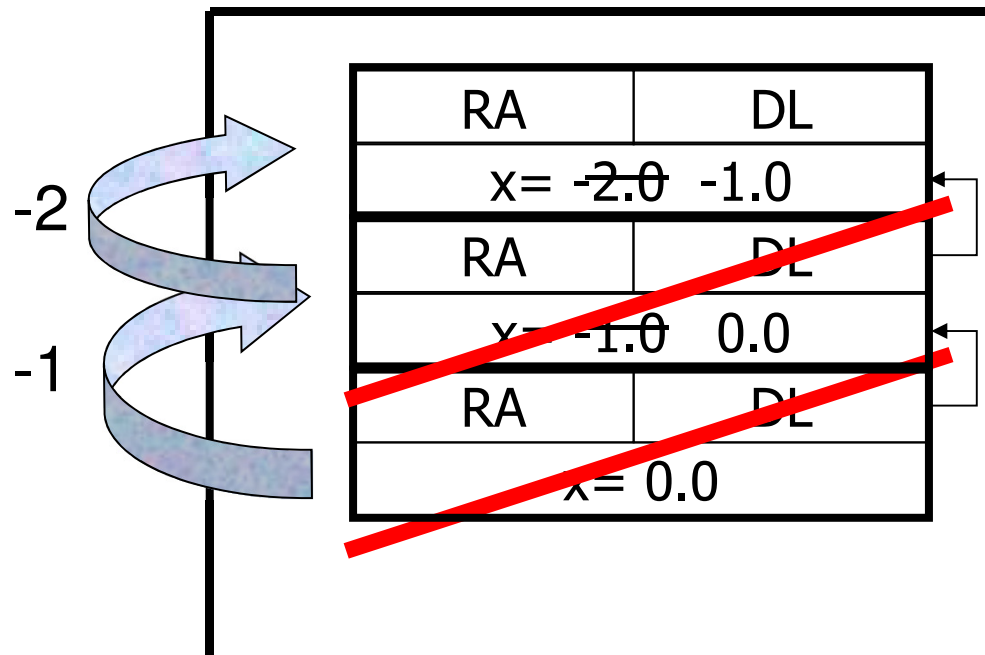
```
double W(int x) {  
  if (x<0) {  
    x++;  
    return W(x)+W(x/2);  
    x++;  
  }  
  else  
    return -1;  
}
```

prima di qua

poi l'eliminazione dell'ultimo record, e ...

Esercizio Record di Attivazione (9) Soluzione

Supponendo che la valutazione degli addendi nella somma venga fatta a partire da sinistra, si ottiene prima:



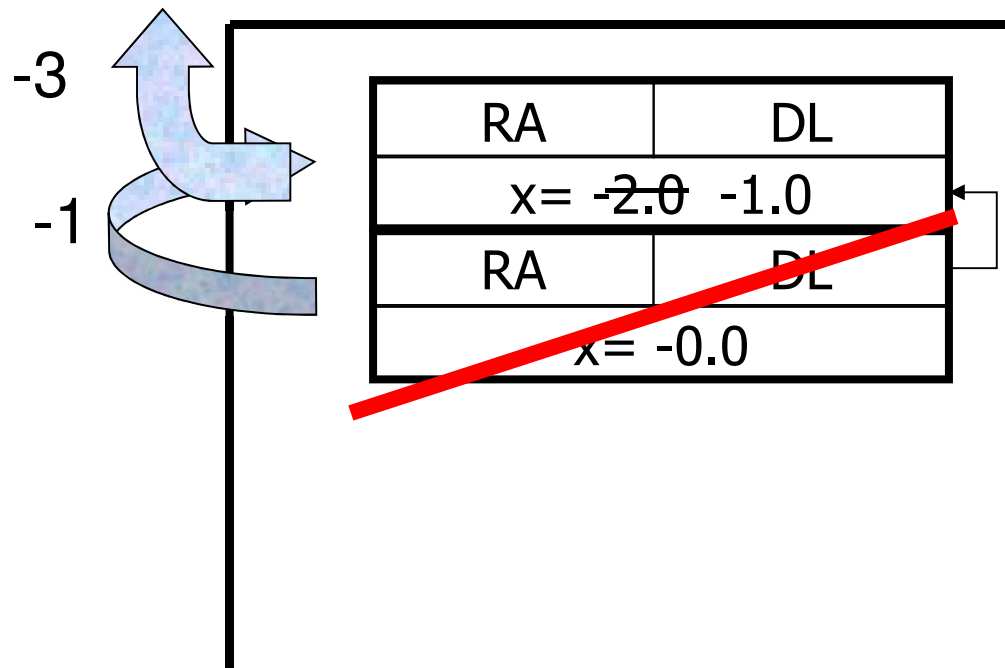
```
double W(int x) {
  if (x < 0) {
    x++;
    return W(x) + W(x/2);  $-1$ ;
    x++;
  }
  else
    return -1;
}
```

ora di qua

Quindi viene fatta la prima somma con risultato -2.00 ,
restituita indietro

Esercizio Record di Attivazione (9) Soluzione

...nel primo record il primo addendo vale -2. Ora dobbiamo calcolare il secondo addendo: $W((\text{int})-1.0/2) \Rightarrow W(0)$



```
double W(int x){  
  if (x<0) {  
    x++;  
    return W(x) + W(x/2); -1;  
    x++;  
  }  
  else  
    return -1;  
}
```

ora di qua

La somma finale (-3) è restituita alla prima invocazione di W.

Esercizio sintesi (10)

Uno dei più antichi sistemi di codificazione di messaggi segreti si basa sulla sostituzione, secondo un certo ordine, dei caratteri componenti il messaggio.

Ad esempio, dato un messaggio composto dalle lettere:

$$\{a, b, c\}$$

E data una chiave di sostituzione che, per ogni lettera ne associa un'altra:

$$'a' \rightarrow 'x'$$
$$'b' \rightarrow 'y'$$
$$'c' \rightarrow 'z'$$

Il messaggio originale può essere così riscritto:

$$\{x, y, z\}$$

Esercizio sintesi (10)

Si vuole costruire un sistema di codifica/decodifica di questo tipo, facendo le seguenti assunzioni:

1. Le lettere componenti il messaggio sono tutte minuscole, ed i messaggi non possono contenere altri caratteri che lettere (no spazi, no numeri)
2. Il codice di sostituzione è dato da un array di 26 caratteri, che viene interpretato nel seguente modo: nella posizione ad indice 0 vi è il carattere che deve sostituire la lettera 'a', in posizione con indice 1 vi è il carattere che deve sostituire la lettera 'b', etc.

Esercizio sintesi (10)

Si strutturi la soluzione implementando due funzioni:

```
void crypt( char source[],  
           int length,  
           char code[DIM_ALPHA],  
           char dest[]);
```

```
void decrypt( char source[],  
            int length,  
            char code[DIM_ALPHA],  
            char dest[]);
```

Ed infine si scriva un semplice main di prova.

Esercizio sintesi (10)

Soluzione

```
void crypt(char source[], int length, char
    code[DIM_ALPHA], char dest[]) {
    int i;

    for (i=0; i<length; i++) {
        dest[i] = code[source[i] - 'a'];
    }
}
```

Esercizio sintesi (10)

Soluzione

```
void decrypt(char source[], int length, char
             code[DIM_ALPHA], char dest[])
{
    int i;
    int j;
    int pos= -1;

    for (i=0; i<length; i++) {
        for (j=0; j<DIM_ALPHA && pos<0; j++) {
            if(source[i] == code[j])
                pos = j;
        }
        dest[i] = 'a' + pos;
        pos = -1;
    }
}
```

Esercizio sintesi (10)

Soluzione

```
#define DIM 256
#define DIM_ALPHA 26

int main()
{
    char source[DIM] = "abc";
    char dest1[DIM] = {'\0', ...};
    char dest2[DIM] = {'\0', ...};
    char codice[DIM_ALPHA] = "cab";

    printf("ORIGINALE: %s\n", source);
    crypt(source, 3, codice, dest1);
    printf("CRIPTATO: %s\n", dest1);
    decrypt(dest1, 3, codice, dest2);
    printf("DE-CRIPTATO: %s\n", dest2);

    return 0; }
}
```