

**Esercizi di preparazione alla I Prova Intermedia di
Fondamenti di Informatica L-A a.a. 2002/2003**

Prof. Paola Mello, Prof. Paolo Bellavista

Il compito comprenderà un esercizio (simile a uno di quelli qui esemplificati) su ciascuno dei seguenti argomenti:

- **ESERCIZI DI ANALISI: Cicli e Funzioni**
- **ESERCIZI DI SINTESI: Cicli e Funzioni**
- **ESERCIZI SU FUNZIONI RICORSIVE e RECORD DI ATTIVAZIONE**
- **DOMANDE (teoria)**
- **ESERCIZI SULLE GRAMMATICHE**
- **ESERCIZI SULLA RAPPRESENTAZIONE DEGLI INTERI**

Non sarà consentito consultare appunti o testi di qualunque tipo durante la prova intermedia

ANALISI: cicli e funzioni

Esercizio

Dato il seguente programma C:

```
#include <stdio.h>

main()
{ int A=0;
  int i;
  for (i=1; i<6; i+=3)
    A = A + i;

  if (A) printf("%d\n", i-A);
    else printf("%d\n", A);
}
```

Qual è l'uscita del programma? La risposta deve essere opportunamente motivata.

Soluzione:

Nel ciclo for nel main si modificano i valori di A con indice $i = 1$ e 4.

A diventa $0 + 1 + 4 = 5$;

i esce con valore 7;

l'istruzione if, essendo A non nulla (valore vero), stampa 7-5 cioè 2.

Esercizio

Si indichino, nel giusto ordine, i valori stampati dal seguente programma C, motivando la risposta data.

```
#include <stdio.h>

void main ()
{
    int i, j;

    for (i=0; i<5; i=i+2)
        { for (j=0; j<3; j++)
            printf(" %d", i+j);
          printf("\n");
        }
}
```

Soluzione:

All'interno del primo ciclo for si assegna il valore 0 a i e di conseguenza si stampa:

0 1 2

successivamente a i si assegna il valore 2 e si stampa:

2 3 4

ancora, si assegna a i il valore 4 e si stampa:

4 5 6

Poi si termina.

Esercizio

Si indichino, nel giusto ordine, i valori stampati dal seguente programma C, motivando la risposta data.

```
#include <stdio.h>

main()
{
    int i,j,x,y;

    printf("Inserire valori interi compresi fra 1
           e 20: ");

    scanf("%d:%d", &x, &y);

    for (i=1; i<=y; i++) {

        for (j=1; j<=x; j++)
            printf("@");

        printf("\n");

    }
}
```

Soluzione:

Dopo aver inserito i due numeri interi separati fra loro da :, il programma stampa un numero di righe pari a y (secondo valore inserito, ciclo for esterno), ciascuna delle quali contiene un numero x (primo valore inserito, ciclo for interno) di caratteri '@'.

In realtà il programma non effettua nessun controllo sul fatto che gli interi inseriti siano compresi nell'intervallo [1, 20].

Esercizio

Dato il seguente programma C:

```
#include <stdio.h>
int Dim=8;

int f(int k)
{int i, s=Dim;
  for(i=Dim-1; i>=0; i-=k)
    s+=i;
  return s;
}

main()
{printf("%d\n", f(3));
}
```

Qual è l'uscita del programma? La risposta deve essere opportunamente motivata.

Soluzione

L'uscita del programma è 20.

La funzione f() viene infatti invocata con valore attuale di k=3.

Il ciclo for viene eseguito con i inizialmente uguale a 7 e decrementato ogni volta di k fino a che i non diventa minore di 0.

Ad ogni ciclo viene sommato ad s il valore corrente di i. La funzione poi restituisce s al programma principale.

Quindi:

$$s = \text{Dim} + (\text{Dim}-1) + (\text{Dim}-1-k) + (\text{Dim}-1-2*k) = 8 + 7 + 4 + 1 = 20$$

ESERCIZI DI SINTESI: Cicli e Funzioni

Esercizio

Si scriva un programma C che legga da terminale una sequenza di numeri interi terminati dal valore 0 e stampi a terminale la media degli interi positivi pari letti.

Soluzione:

```
#include <stdio.h>

main()
{
    float media;
    int num, n_pos_pari;

    media=0; n_pos_pari=0;
    do {
        printf("Inserisci un numero (0 per
            terminare)\n");
        scanf("%d", &num);
        if (num>0)
            if (num%2==0)
                {media=media+num;
                n_pos_pari++;
                }
    }
    while (num!=0);

    media = media/n_pos_pari;
    printf("La media vale: %f\n", media);
}
```

Esercizio

Si realizzi un programma C che legga da standard input i dati relativi ai corsi tenuti presso un'università. In particolare, l'utente inserisce per ogni corso (si supponga che il numero dei corsi sia inserito dall'utente all'inizio della computazione) un intero che indica il numero di studenti che frequentano il corso ed un altro intero che ne indica il codice.

Il programma deve stampare il codice di tutti i corsi che hanno il numero di iscritti maggiore o uguale al valore 30.

Soluzione

```
#include <stdio.h>

main()
{
    int isc, corso, i, nc;
    printf("Inserisci il numero dei corsi: ");
    scanf("%d", &nc);

    /* inserimento dati */
    for (i=0; i<nc; i++)
    {
        printf("iscritti e codice corso");
        scanf("%d%d", &isc, &corso);
        if (isc>=30)
            printf("Corso con più di 30
                    iscritti: %d\n", corso);
    }
}
```

Esercizio

Si scriva una funzione **ricorsiva**
double f(double a, int n);
che calcoli il seguente valore:

$$\sum_{i=1}^n (a - i/a)$$

Soluzione

```
double f(double a, int n)
{ if (n==1) return a - 1/a;
  else return a - n/a + f(a, n-1);
}
```


Esercizio

Si scriva la versione iterativa (utilizzo di **while**) della funzione:

```
double f(double a, int n);
```

che calcola il seguente valore

$$\sum_{i=1}^n (a - i/a)$$

Soluzione

```
double f(double a, int n)
{ int i=1;
  double sum=0;
  while(i<=n)
    {sum = sum + a - i/a;
     i++;}
  return sum;
}
```

Esercizio

Si scriva una funzione **ricorsiva**
`double f(int m, double n);`
che calcoli il seguente valore

$$\prod_{i=1}^m n/i$$

Soluzione

```
double f(int m, double n)
{ if (m==1) return n;
  else return n/m * f(m-1,n);
}
```

Esercizio

Si scriva una funzione iterativa (ciclo **while**):

```
double f(int m, double n);
```

che calcoli il seguente valore

$$\prod_{i=1}^m n/i$$

Soluzione

```
double f(int m, double n)
{ int i=1;
  double prod=1;
  while (i<=m)
    { prod = prod * (n/i); i++;
    }
  return prod;
}
```

Esercizio: Numeri Primi

Si scriva una funzione C che calcola se un numero è primo, utilizzando la seguente specifica definita ricorsivamente:

- 2 è primo.
- un numero N è primo se non è divisibile per alcun numero primo strettamente compreso fra 2 ed $N/2$.

Soluzione:

```
int div(int N,int M)
{ return (N % M) == 0;
}

int primo(int N)
{ int i, Trovato = 1;
  if (N==2) return 1;
  else{
    for (i=2; i<N/2; i++)
      if (primo(i))
        Trovato = Trovato && !div(N, i);
    return Trovato;
  }
}
```

ESERCIZI su FUNZIONI RICORSIVE e RECORD di ATTIVAZIONE

Esercizio

Si consideri la seguente funzione F la cui specifica è data in modo ricorsivo (si supponga N intero):

$F(N) =$ restituisce 1 se $N \leq 0$,
 $-1+4*F(N-1) + F(N-2)$, altrimenti

1. Si scriva la funzione C che realizza tale specifica;
2. Si scriva il risultato della funzione quando chiamata con $N=3$ e si mostri la sequenza dei record di attivazione;
3. Si scriva il programma main che chiama tale funzione con $N=3$ e ne stampa il risultato a terminale.

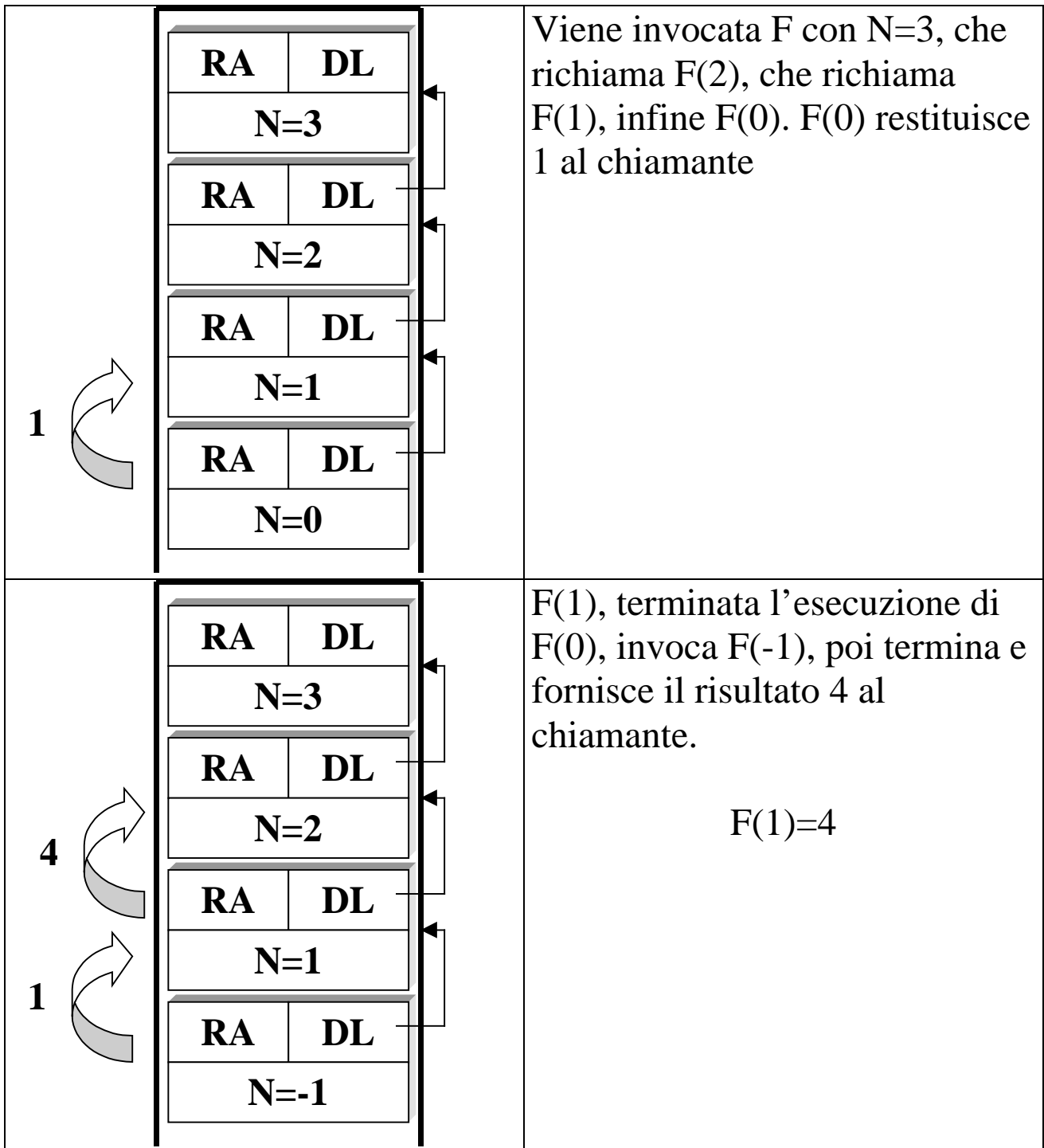
Soluzione

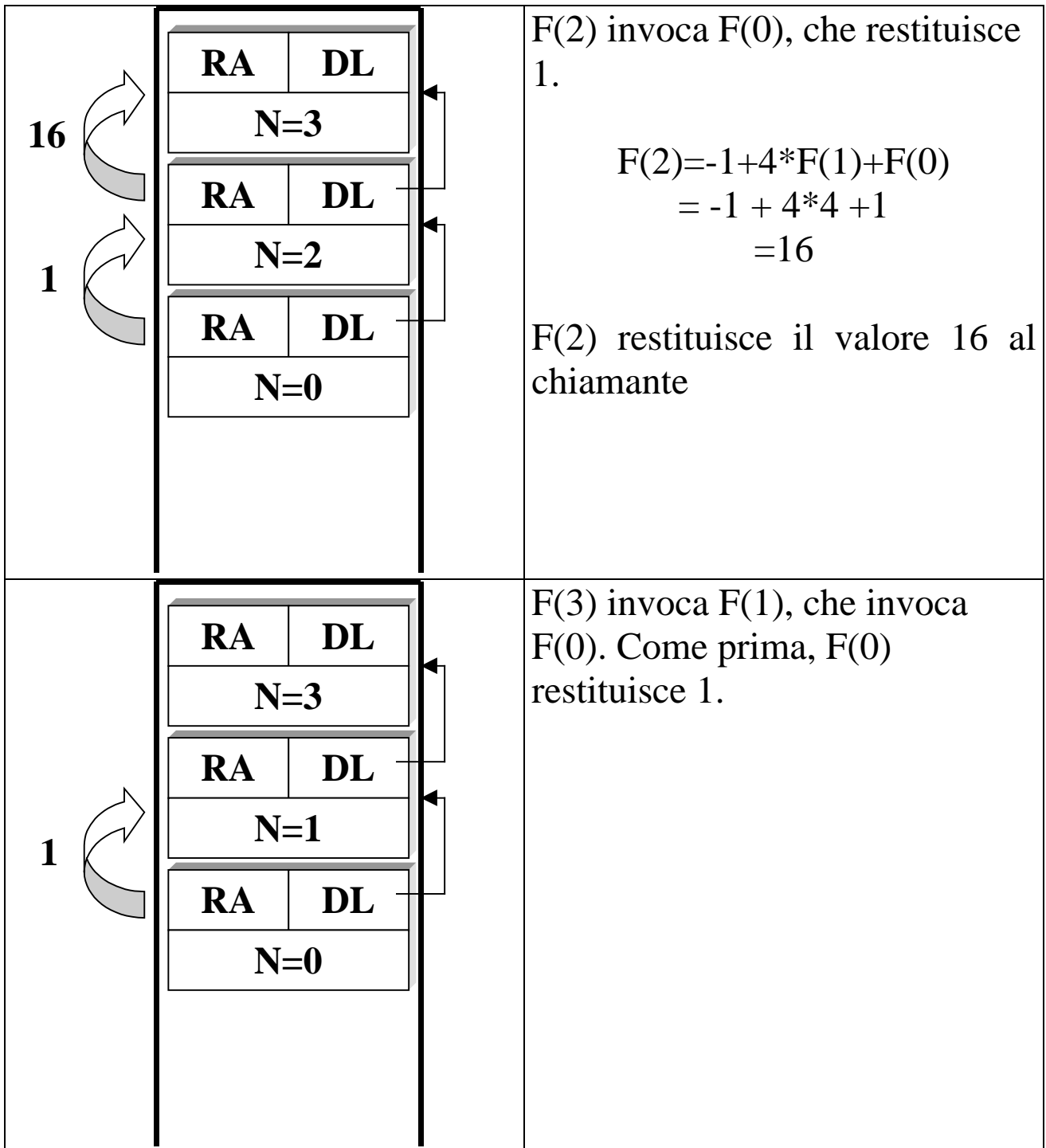
```
int F(int N)
{
    if (N<=0) return 1;
    else return -1+4*F(N-1) + F(N-2);
}
main (){
printf("risultato %d", F(3)) ;}
```

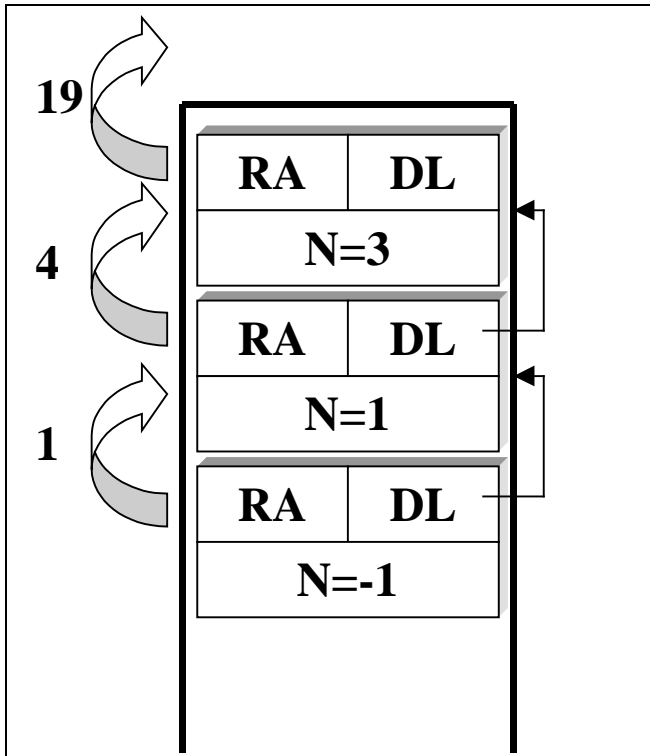
oppure:

```
int F(int N);
main (){
printf("risultato %d", F(3)) ;}

int F(int N)
{
    if (N<=0) return 1;
    else return -1+4*F(N-1) + F(N-2);
}
```







F(1) invoca F(-1), che restituisce 1. F(1) termina restituendo 4 al chiamante F(3)).

$$\begin{aligned}
 F(3) &= -1 + 4 * F(2) + F(1) \\
 &= -1 + 4 * 16 + 4 \\
 &= 67
 \end{aligned}$$

F(3) termina e restituisce 67

Esercizio

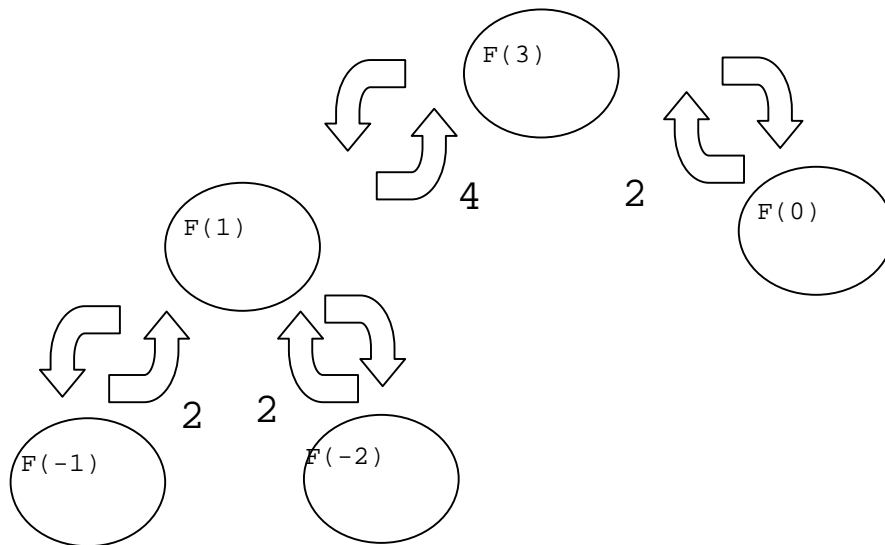
Si consideri la seguente funzione F la cui specifica è data in modo ricorsivo (si supponga N intero):

$F(N) =$ restituisce 2 se $N \leq 0$,
 $F(N-2) * F(N-3)$, altrimenti

- Si scriva il risultato della funzione quando chiamata con $N=3$ e si mostrino i valori intermedi assunti da N ;
- Si scriva la funzione C che realizzerebbe tale specifica
- Si mostrino i record di attivazione nello stack

Soluzione:

Sequenza di attivazioni:

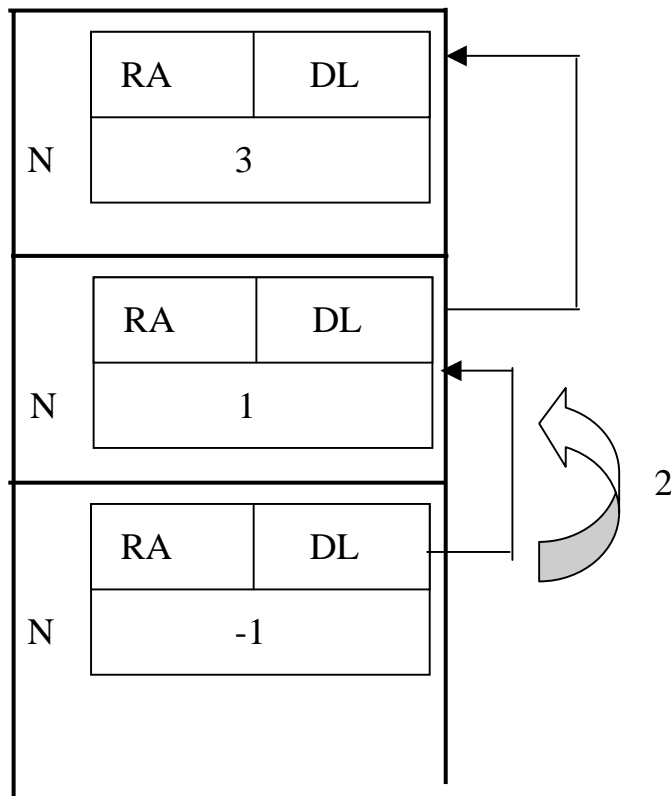


$F(3)$ restituisce il valore 8

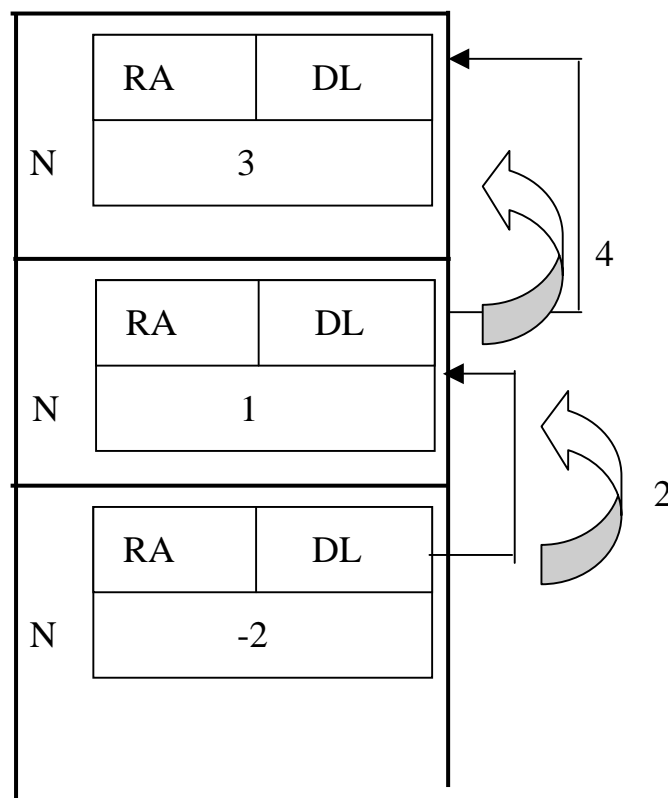
Valori assunti da N : 3 1 -1 -2 0

```
int F(int N)
{if (N<=0) return 2;
 else return F(N-2)*F(N-3);
}
```

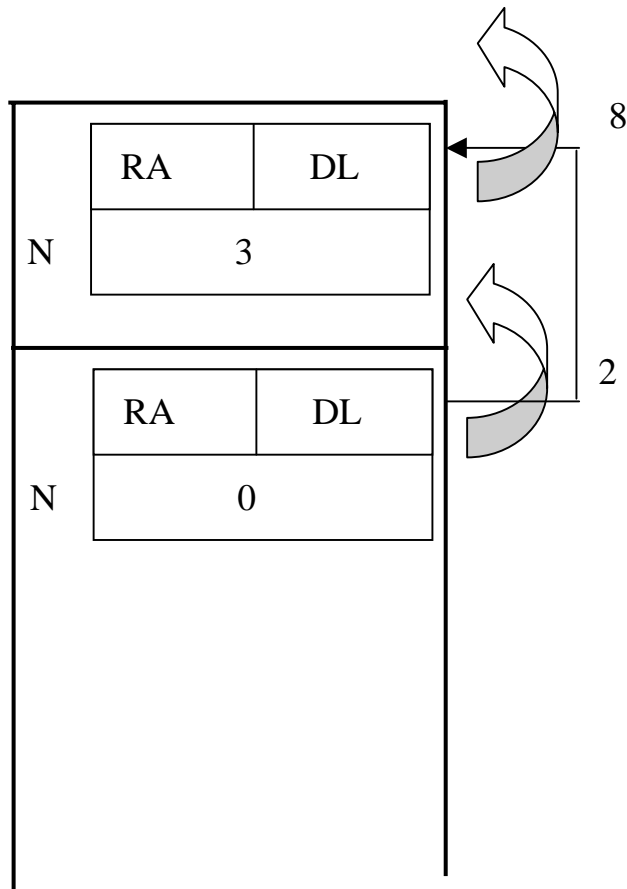
Record di attivazione della chiamata con $N=3$



La $F(3)$ chiama la $F(1)$ che a sua volta chiama $F(-1)$. Quest'ultima finisce restituendo il valore 2



Dopo di che la funzione $F(1)$ chiama $F(-2)$. Quest'ultima finisce restituendo il valore 2. Quindi finisce anche $F(1)$ restituendo il valore 4 (risultato della moltiplicazione tra 2 e 2).



Dopo di che la funzione $F(3)$ chiama $F(0)$. Quest'ultima finisce restituendo il valore 2. Quindi finisce anche $F(3)$ restituendo il valore 8 (risultato della moltiplicazione tra 4 e 2).

Esercizio

Si consideri la seguente funzione P la cui specifica è data in modo ricorsivo (si supponga N intero):

$P(N) =$ stampa N se $N \leq 10$,
stampa N e invoca $P(N-10)$, altrimenti

- Scrivere il codice C di tale funzione;
- La funzione è tail ricorsiva?
- Si scriva la sequenza di valori stampati quando la funzione è chiamata con $N = 30$. Si mostri anche la sequenza dei record di attivazione, sia nel caso il compilatore effettui l'ottimizzazione tail, sia se non la effettua.

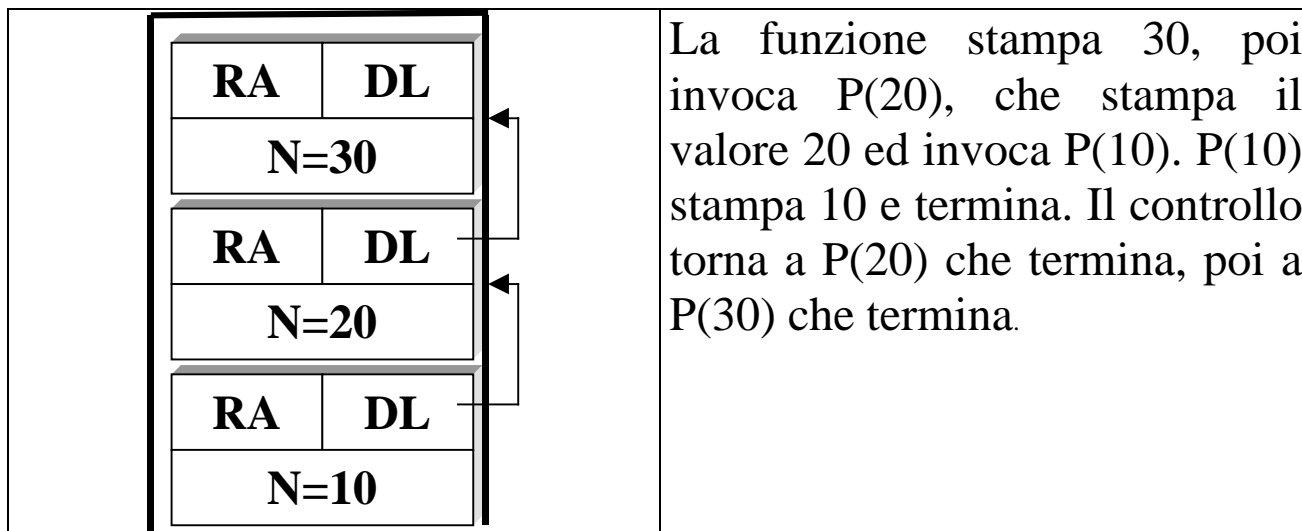
Soluzione:

- Codice della funzione

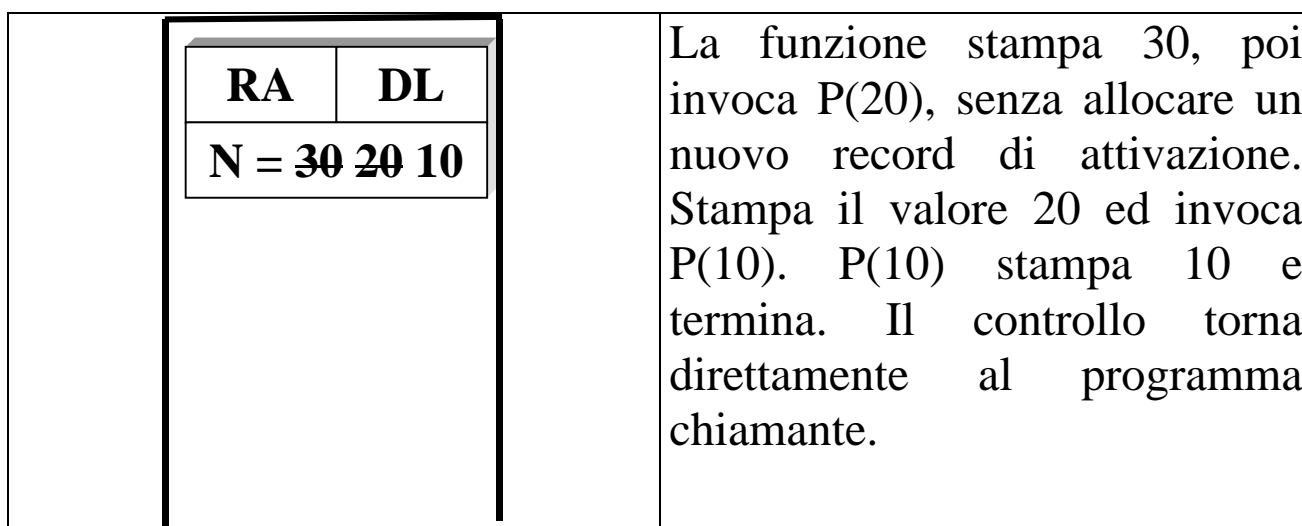
```
void P(int N)
{ printf("%d ", N);
  if (N>10)
    { P(N-10); }
}
```

- La funzione è tail ricorsiva, in quanto la chiamata ricorsiva è l'ultima operazione effettuata.

- La funzione stampa la sequenza
30 20 10



Se il compilatore effettua l'ottimizzazione tail, viene riutilizzato sempre lo stesso record di attivazione:



DOMANDE (teoria)

Esercizio

Qual è la differenza tra un processo computazionale ricorsivo e uno ricorsivo tail?

- A. Il primo può essere molto più efficiente del secondo perché nel secondo caso vengono allocati tanti record di attivazione quante sono le chiamate, nel primo uno solo.
- B. Il secondo può essere molto più efficiente del primo perché nel secondo caso si può effettuare un'ottimizzazione che limita il numero di record di attivazione allocati sullo stack.
- C. Nessuna differenza.

Soluzione

La risposta corretta è la B.

Esercizio

Che cosa accade se all'interno di una funzione/procedura si modifica un parametro passato per riferimento?

- A. Si agisce su una copia e quindi il parametro attuale del chiamante non viene modificato.
- B. Si ottiene un errore di compilazione.
- C. Viene modificato il valore della variabile passata come parametro attuale dal chiamante.

Soluzione:

La risposta corretta è la C.

Esercizio

Qual è la differenza tra un parametro formale passato per indirizzo/riferimento e uno passato per valore ad una procedura?

- A. Se modificato all'interno della procedura, il parametro passato per indirizzo non comporta modifica sul parametro attuale. Viceversa avviene per il parametro passato per valore.
- B. Se modificato all'interno della procedura, il parametro passato per valore non comporta modifica sul parametro attuale. Viceversa avviene per il parametro passato per indirizzo/riferimento.
- C. I parametri passati per indirizzo sono solo vettori, mentre tutti gli altri non possono essere passati che per valore.

Soluzione:

La risposta corretta è la B.

Esercizio

Identificare e correggere gli errori in ognuna delle seguenti istruzioni:

- **while (c<=5) {p *= c; ++c;**

(errore sintattico manca una parentesi graffa alla fine del **while**)

- **while (c<=5) {p = p * c; } ++c;**

(errore logico, **c** non è modificata all'interno del ciclo, loop infinito)

Esercizio

Si dia una definizione sintetica di funzione computabile:

Soluzione:

È una funzione f per la quale esiste una Macchina di Turing che data sul nastro una rappresentazione di **un elemento appartenente al suo dominio x** dopo un numero finito di passi produce sul nastro una rappresentazione del risultato **$f(x)$** .

ESERCIZI SULLE GRAMMATICHE

Esercizio

Si consideri la grammatica $G = (V_n, V_t, P, S)$, con
 $V_n = \{ \langle U \rangle, \langle D \rangle \}$, $V_t = \{ 1, 2, 3 \}$, $S = \langle U \rangle$ e

$$P = \left\{ \begin{array}{l} \langle U \rangle ::= 1 \langle D \rangle 3 \\ \langle D \rangle ::= 2 \langle U \rangle \mid 2 \\ \end{array} \right\}$$

- Si dica se la stringa 121233 è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione left-most.
- Si dica se la stringa 121233 è sintatticamente corretta rispetto a tale grammatica e se ne mostri l'albero sintattico.

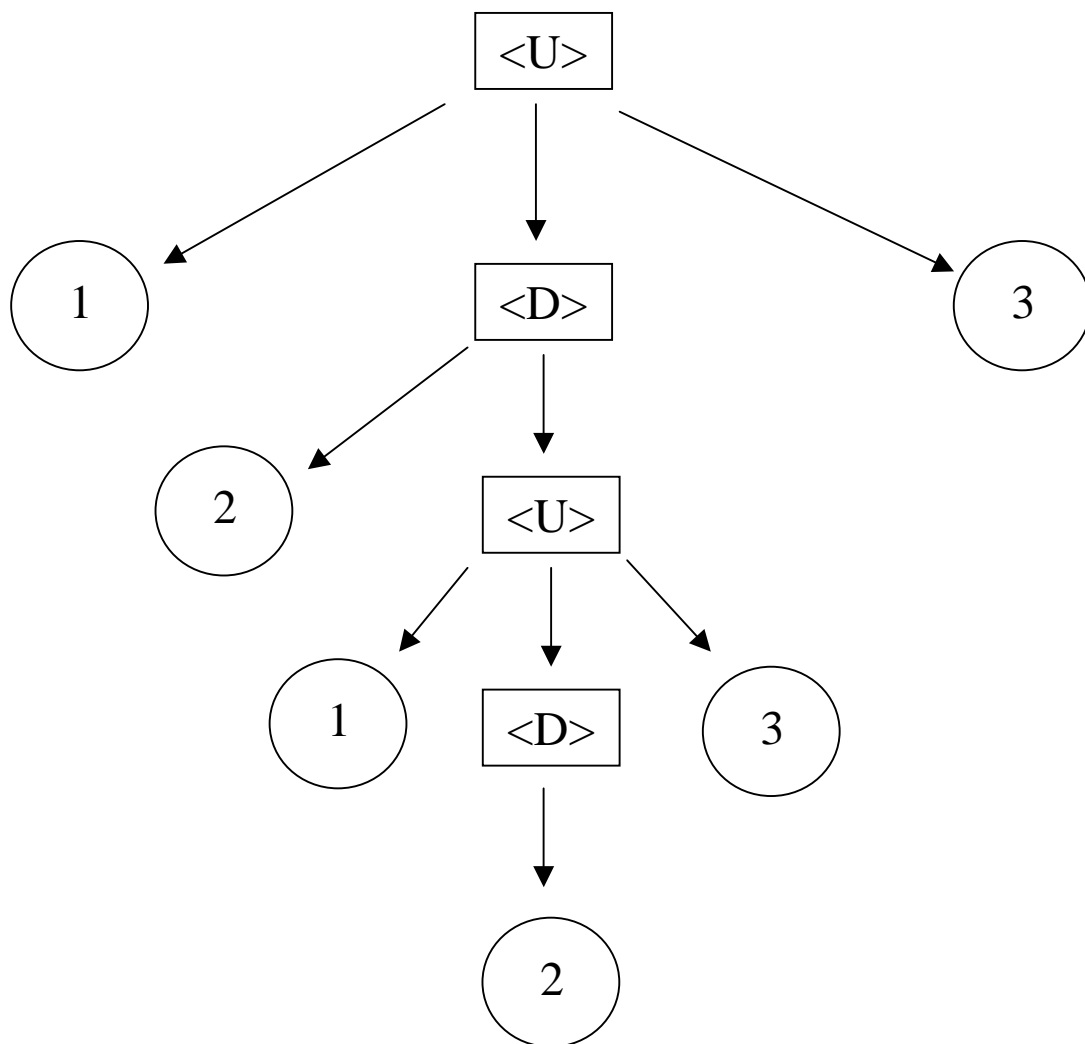
Soluzione:

Derivazione left-most

$\langle U \rangle$

- ➔ 1 $\langle D \rangle$ 3
- ➔ 1 2 $\langle U \rangle$ 3
- ➔ 1 2 1 $\langle D \rangle$ 3 3
- ➔ 1 2 1 2 3 3

Albero sintattico



Esercizio

Data la grammatica

$G = \{VN, VT, S, P\},$

$VN = \{\langle T \rangle, \langle F \rangle\},$

$S = \langle T \rangle$

$VT = \{\text{and, not, true, false}\},$

$P = \{$

$\langle T \rangle ::= \langle F \rangle \text{ and } \langle T \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \text{not } \langle F \rangle \mid \text{true} \mid \text{false}$

$\}$

- Dire se la stringa
not true and false and not false
è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione left-most.
- Dire se la stringa
not true and false and not false
è sintatticamente corretta rispetto a tale grammatica e se ne mostri l'albero sintattico.

Soluzione:

Derivazione left-most

$\langle T \rangle$

→ $\langle F \rangle \text{ and } \langle T \rangle$

→ $\text{not } \langle F \rangle \text{ and } \langle T \rangle$

→ $\text{not true and } \langle T \rangle$

→ $\text{not true and } \langle F \rangle \text{ and } \langle T \rangle$

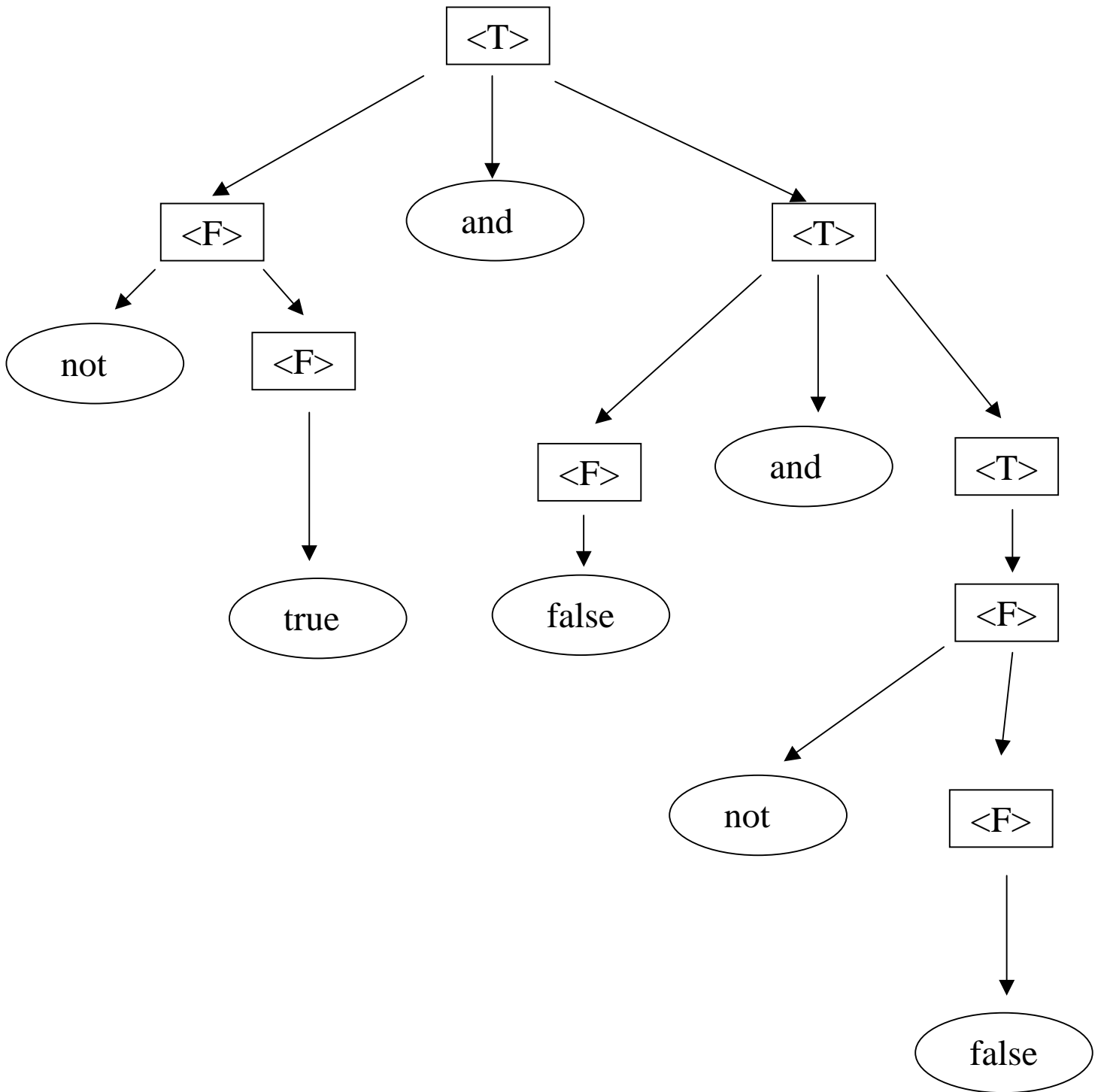
→ $\text{not true and false and } \langle T \rangle$

→ $\text{not true and false and } \langle F \rangle$

→ $\text{not true and false and not } \langle F \rangle$

→ $\text{not true and false and not false}$

Albero sintattico



Esercizio

Data la grammatica G con scopo S e simboli terminali $\{a, b, c, d, 0, 1\}$

$S ::= Aa \mid bB \mid bC$

$A ::= 1B \mid 0$

$B ::= 0A \mid 1$

$C ::= Cc \mid d$

Si mostri (mediante derivazione left-most) che la stringa **b011** appartiene alla grammatica, mentre la stringa **b111** no.

Soluzione:

Derivazione left-most della stringa b011

S

→ bB

→ $b0A$

→ $b01B$

→ $b011$

Derivazione left-most della stringa b111 (non possibile)

S

→ bB

→ $b1$

e non è più possibile proseguire

Esercizio

Si consideri la grammatica $G = (V_n, V_t, P, S)$, con $V_n = \{U, Z, T\}$, $V_t = \{0, 1, 2, 3\}$, $S = U$ e

$$P = \left\{ \begin{array}{l|l} U ::= 3T & 1Z \\ Z ::= 3T & 0U \\ T ::= 2 & \end{array} \right. \}$$

Si mostri (mediante derivazione left-most) che la stringa **10132** appartiene alla grammatica, mentre la stringa **130** no.

Derivazione left-most della stringa 10132

U
→ 1Z
→ 10U
→ 101Z
→ 1013T
→ 10132

Derivazione left-most della stringa 130 (non possibile)

U
→ 1Z
→ 13T
→ ??

non esiste

ESERCIZI sulla RAPPRESENTAZIONE dei NUMERI NATURALI/INTERI e OPERAZIONI di SOMMA e SOTTRAZIONE

ESERCIZIO 1

Effettuare le seguenti operazioni aritmetiche tra numeri binari naturali, ipotizzando di lavorare con un elaboratore con lunghezza di parola (word) pari a un byte:

- Differenza: $A-B$, $A=(35)_{10}$, $B=(22)_{10}$

$$(35)_{10} = (23)_{16} = (0010\ 0011)_2$$

$$(22)_{10} = (16)_{16} = (0001\ 0110)_2$$

$$0010\ 0011 -$$

$$0001\ 0110 =$$

$$0000\ 1101 \quad \text{Prestito: 0}$$

Risultato: $A - B = (0000\ 1101)_2 = (0D)_{16} = (13)_{10}$

- Somma: $A+B$, $A=(42)_{10}$, $B=(31)_{10}$

$$(42)_{10} = (2A)_{16} = (0010\ 1010)_2$$

$$(31)_{10} = (1F)_{16} = (0001\ 1111)_2$$

$$0010\ 1010 +$$

$$0001\ 1111 =$$

$$0100\ 1001$$

Risultato: $A + B = (0100\ 1001)_2 = (49)_{16} = (73)_{10}$

ESERCIZIO

Un elaboratore rappresenta numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica:

$$59 + 250$$

in codifica binaria e se si commettono errori.

Soluzione

$$+59 \rightarrow 0 \ 0111011$$

+250 non è rappresentabile in modulo e segno con soli 8 bit.

ESERCIZIO

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica:

$$96-17$$

in codifica binaria. Trasformare il risultato binario finale in decimale, per verifica.

Soluzione

$$96 \rightarrow 0 \ 1100000$$

$$-17 \rightarrow 1 \ 0010001$$

Tra i (moduli dei) due numeri si esegue una sottrazione:

$$\begin{array}{r} 1100000 \\ - 0010001 \\ \hline 1001111 \end{array}$$

che vale 79 in base 10.

Un esempio di possibile compito intermedio

Fondamenti di Informatica L-A (a.a. 2001/2002) - Ingegneria Informatica – prof.^{ssa} Mello
Testo della prova intermedia del 9/11/2001

COMPITO A

▪ Esercizio 1

Dato il seguente programma C:

```
#include <stdio.h>

int f1(int x, int y) {
    if (x>y) return x;
    else return y;}

main() {
    int x=0, z=0;
    for (x=5; x>0; x-=2) {
        z=f1(x, z);
        printf("%d\n", x+z);
    }
}
```

Che cosa fa la funzione f1? Cosa stampa il programma (motivare opportunamente la risposta)?

▪ Esercizio 2

Si scriva una funzione **ricorsiva** `int f(int m, int c)`; che calcoli il seguente valore:

$$\prod_{i=1}^m c*(i+i)$$

▪ Esercizio 3

Si consideri la seguente funzione F:

```
int F(int a){
    if (a<2) return 1;
    else return 3*F(a-2);}
```

Si scriva il risultato della funzione quando invocata come F(5) e si mostrino i record di attivazione. La funzione è tail ricorsiva ?

▪ Esercizio 4

Data la grammatica **G** con scopo **S** e simboli terminali **{a,c,0,1}**

S ::= a F c

F ::= a S c | E

E ::= 0 | 1

si mostri (mediante derivazione left-most) che la stringa **aaa1ccc** appartiene alla grammatica.

▪ Esercizio 5

Si trovi l'errore logico nelle seguenti istruzioni che dovrebbero visualizzare gli interi dispari da 99 a 3 e si presenti la versione corretta

```
for (x=99; x>=3; x+=2) printf("%d\n", x);
```

▪ Esercizio 6

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

Soluzioni

Esercizio 1

Il programma stampa a video la sequenza di interi, ciascuno su una linea diversa:

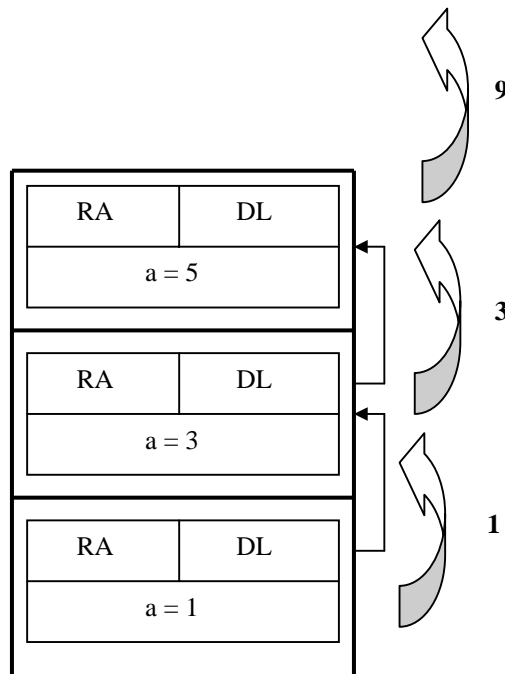
10 8 6

Infatti, la prima esecuzione del ciclo avviene con $x=5$. La funzione $f1()$ non fa altro che restituire il maggiore fra x e z , e il risultato viene assegnato a z ($z=5$). La $\text{printf}()$ stampa la somma ($x+z$), cioè 10. La seconda esecuzione del ciclo avviene con $x=3$; z assume il valore maggiore fra 3 e 5, ossia 5, e viene stampato a video 8. L'ultima esecuzione del ciclo avviene con $x=1$; il massimo tra 1 e 5 è ancora 5 e a video viene stampato 6. La variabile x viene poi decrementata di due unità e la condizione $x>0$ non è più verificata, quindi il corpo del ciclo non è eseguito e il programma termina.

Esercizio 2

```
int f(int m, int c)
{ if (m==1) return 2*c;
  else return c*(m+m) * f(m-1);
}
```

Esercizio 3



La funzione non è tail ricorsiva, poiché l'ultima istruzione eseguita dalla funzione è il prodotto tra 3 e il valore restituito dalla chiamata ricorsiva.

Esercizio 4

S → aFc → aaSc → aaFccc → aaEccc → aa1ccc

Esercizio 5

Il ciclo non termina. Versione corretta:

```
for (x=99; x>=3; x-=2) printf("%d\n",x);
```

Esercizio 6

112 -> 0 1110000

-14 -> 1 0001110

Tra i moduli dei numeri si esegue una sottrazione:

```
  1110000
-   0001110
-----
  1100010
```

che vale 98 in base dieci.