

Esercizi di preparazione allo scritto

Esercizio 1

Sia dato il file binario "pazienti.bin" contenente i dati relativi a tutti i pazienti di un ospedale. In particolare, ogni record del file contiene i dati relativi ad un paziente ed ha la seguente struttura:

- *Nome*: una stringa che rappresenta il nome del paziente;
- *Cognome*: una stringa che rappresenta il cognome del paziente;
- *Tessera*: un intero che rappresenta il numero di tessera sanitaria del paziente e che lo individua univocamente.

I pazienti possono appartenere a due categorie: *ricoverati* (che sono attualmente ricoverati nell'ospedale) e *dimessi* (ex-ricoverati, attualmente dimessi).

A questo scopo, sia dato un secondo file di testo "*dentro.txt*" che riporta l'elenco dei pazienti ricoverati. In particolare, ogni linea del file contiene il numero di tessera di un paziente ricoverato.

Esercizio di preparazione allo scritto

Domande:

Si scriva un programma in linguaggio C che:

- 1 .A partire dai file "pazienti.bin" e "dentro.txt" costruisca due tabelle TR e TD contenenti rispettivamente Nome, Cognome e Tessera dei pazienti ricoverati (tabella TR) e dimessi (tabella TD).
- 2 .A partire dalle tabelle TR e TD, e dato da input un intero N stampi cognome e nome dei pazienti il cui numero di tessera è minore di N.
- 3 .A partire dalle tabelle TR e TD, scriva su un terzo file di testo "giuseppe.txt", Nome, Cognome e numero di tessera di tutti i pazienti con nome uguale a "giuseppe".

Progetto dei dati:

Tipo di dato contenuto nel file pazienti.bin:

```
typedef struct {      char nome[20];
                      char cognome[30];
                      int tessera;
                    } paziente;
```

Progetto delle tabelle:

```
typedef  paziente  tabella[100];
tabella TR, TD; /*tabelle da costruire
                 con i dati estratti
                 dai file */
```

Programma:

```
#include <stdio.h>
#include <string.h>
typedef struct {  char nome[20];
                  char cognome[30];
                  int tessera;
                } paziente;

typedef  paziente tabella[10];

main()
{  paziente P;
   tabella TR, TD;
   char nomep[20]= "giuseppe";
   FILE *f1, *f2;
   int i, trovato, insR=0, insD=0, tess;
```

```

/*domanda 1: costruzione delle tabelle */
f1=fopen("pazienti.bin", "rb");
f2=fopen("dentro.txt", "r");
while (fread(&P,sizeof(paziente),1,f1)>0)
{
    trovato=0; rewind(f2);
    while(fscanf(f2,"%d", &tess)>0 && !trovato)
        if (P.tessera==tess) /*P e` ricoverato*/
        {
            trovato=1;
            TR[insR]=P;
            insR++;
        }
    if (!trovato) /*P non e` ricoverato*/
    {
        TD[insD]=P;
        insD++;
    }
}
fclose(f1); fclose(f2);

```

/*domanda 2*/

```
fflush(stdin);
printf("\nInserire il numero N: ");
scanf("%d", &tess);
for(i=0; i<insR; i++)
    if(TR[i].tessera<tess)
        printf("\n%20s\t%30s\n",TR[i].nome,
                TR[i].cognome);
for(i=0; i<insD; i++)
    if(TD[i].tessera<tess)
        printf("\n%20s\t%30s\n",TD[i].nome,
                TD[i].cognome);
```

```
/*domanda 3: creazione file "giuseppe.txt" */  
f1=fopen("giuseppe.txt", "w");  
for (i=0; i<insR; i++)  
    if (strcmp(nomep, TR[i].nome)==0)  
        fprintf(f1, "%s\t%s\t%d\n", TR[i].nome,  
                TR[i].cognome, TR[i].tessera);  
for (i=0; i<insD; i++)  
    if (strcmp(nomep, TD[i].nome)==0)  
        fprintf(f1, "%s\t%s\t%d\n", TD[i].nome,  
                TD[i].cognome, TR[i].tessera);  
fclose(f1);  
}/* fine main*/
```

Esercizio 2

Sia dato il file di testo "*studenti.txt*" contenente i dati relativi agli studenti di un corso universitario. Ogni linea del file contiene i dati relativi ad uno studente; in particolare:

- ***Nome***: una stringa che rappresenta il nome dello studente.
- ***Cognome***: una stringa che rappresenta il cognome del studente.
- ***Matricola***: un intero che individua univocamente lo studente.

Un secondo file binario "*verifiche.bin*" riporta i dati relativi alle 3 prove di verifica intermedie effettuate durante lo svolgimento del corso.

In particolare, ogni record del file rappresenta l'esito di una prova parziale :

- ***Matricola***: un intero che individua univocamente lo studente.
- ***Voto***: un intero che rappresenta il voto riportato nella prova parziale dallo studente.

Domande:

Si scriva un programma in linguaggio C che:

- 1 A partire dai file "studenti.txt" e "verifiche.bin" costruisca una tabella T che riassume le informazioni relative ad ogni studente del corso: in particolare, ogni elemento della tabella rappresenterà uno studente e conterrà quindi *Nome*, *Cognome*, *Matricola*, e *VotoTotale* (cioè, la somma dei voti ottenuti nelle prove parziali) associati allo studente.
- 2 A partire dalla tabella T, e dato da input un voto V stampi nome e cognome di tutti gli studenti che hanno ottenuto un *VotoTotale* maggiore o uguale a V.
- 3 Calcoli la media M di tutti i *VotiTotali* riportati in T e scriva su un terzo file binario "*bravi.bin*", *Nome*, *Cognome*, *Matricola* di tutti gli studenti che hanno ottenuto un *Voto Totale* maggiore o uguale a M.

Progetto dei dati:

Tipo di dato studente (file studenti.txt):

```
typedef struct {      char nome[20];
                      char cognome[30];
                      int  matr;
                    } studente;
```

Tipo di dato esito (file verifiche.bin):

```
typedef struct {      int  matr;
                      unsigned int voto;
                    }esito;
```

Progetto dei dati:

Elemento della tabella:

```
typedef struct {      char nome[20];  
                    char cognome[30];  
                    int matr;  
                    unsigned int vtot;  
                }elemento;
```

Progetto della tabella:

```
typedef elemento tabella[100];
```

Programma:

```
#include <stdio.h>
#include <string.h>
typedef struct {  char nome[20];
                  char cognome[30];
                  int matr;
                } studente;
typedef struct{   int matr;
                 unsigned int voto;
                }esito;
typedef struct{  char nome[20];
                 char cognome[30];
                 int matr;
                 unsigned int vtot;
                }elemento;
typedef elemento tabella[10];
```

```
main()
{ studente s;
  esito e;
  elemento el;
  tabella T;
  FILE *f1, *f2;
  int i,ins=0;
  unsigned int V;
  float m=0;

/*continua...*/
```

```
/*domanda 1: costruzione della tabella */
f1=fopen("studenti.txt", "r");
f2=fopen("verifiche.bin", "rb");
i=0;
while(fscanf(f1,"%s%s%d",&T[i].nome, &T[i].cognome,
            &T[i].matr)>0)
{
    T[i].vtot=0; rewind(f2);
    while(fread(&e, sizeof(esito),1,f2)>0)
        if (e.matr==T[i].matr)
            T[i].vtot+=e.voto;
    i++;
}
fclose(f1);
fclose(f2);
ins=i;
```

```
/*domanda 2*/  
fflush(stdin);  
printf("\nInserire il Voto V: ");  
scanf("%u", &V);  
for(i=0; i<ins; i++)  
    if(T[i].vtot>=V)  
        printf("\n%20s\t%30s\n",T[i].nome,T[i].cognome);  
  
/*domanda 3 */  
/* Calcolo del voto medio: */  
for(i=0; i<ins; i++)  
    m+=T[i].vtot;  
m=m/ins;
```

```
/* scrittura del file "bravi.bin": */  
  
f1=fopen("bravi.bin", "wb");  
  
for (i=0; i<ins; i++)  
    if (T[i].vtot>=m)  
        { strcpy(s.nome, T[i].nome);  
          strcpy(s.cognome, T[i].cognome);  
          s.matr=T[i].matr;  
          fwrite(&s, sizeof(studente), 1, f1);  
        }  
    fclose(f1);  
}/* fine main */
```

3. Esercizio “tipo” per prova scritta: analisi

Testo:

Dato il seguente programma (v. pagina seguente)

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

Programma da analizzare

```
#include <stdio.h>
int f(int a, int b)
{ static int X=0;
  if (!(a||b))
    return 1;
  else    return X=f(a-2,b-1) + X;
}

main()
{ printf("%d\n", f(4,2)); }
```

Stack

Codice

```
int f(int a, int b)
{static int X=0;
if (!(a||b))
    return 1;
else
return X=f(a-2,b-1) + X;
}

main()
{
printf("%d\n", f(4,2));
}
```

main

RA

S.O.

Data Segment:

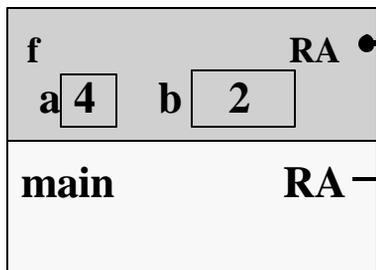
X 0

Stack

Codice

```
int f(int a, int b)
{static int X=0;
  if (!(a||b))
    return 1;
  else
    return X=f(a-2,b-1) + X;
}

main()
{
  printf("%d\n", f(4,2));
}
```



S.O.

Data Segment:

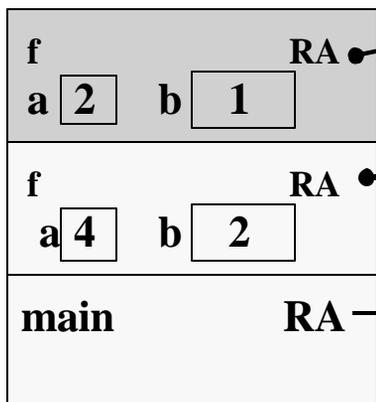
X 0

Stack

Codice

```
int f(int a, int b)
{static int X=0;
  if (!(a||b))
    return 1;
  else
    return X=f(a-2,b-1) + X;
}

main()
{
  printf("%d\n", f(4,2));
}
```



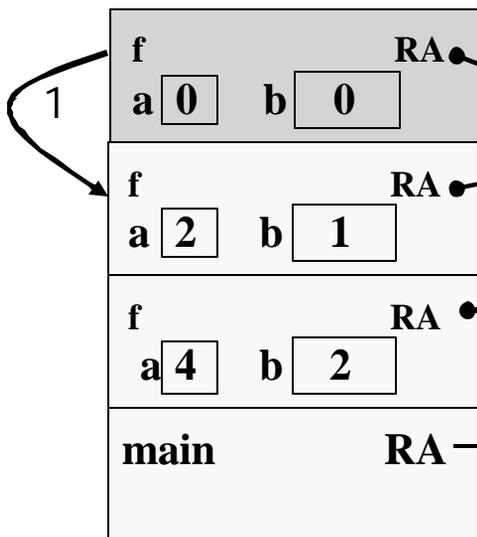
S.O.

Data Segment:

X 0

Stack

Codice



```
int f(int a, int b)
{static int X=0;
if (!(a||b))
    return 1;
else
return X=f(a-2,b-1) + X;
}

main()
{
printf("%d\n", f(4,2));
}
```

S.O.

Data Segment:

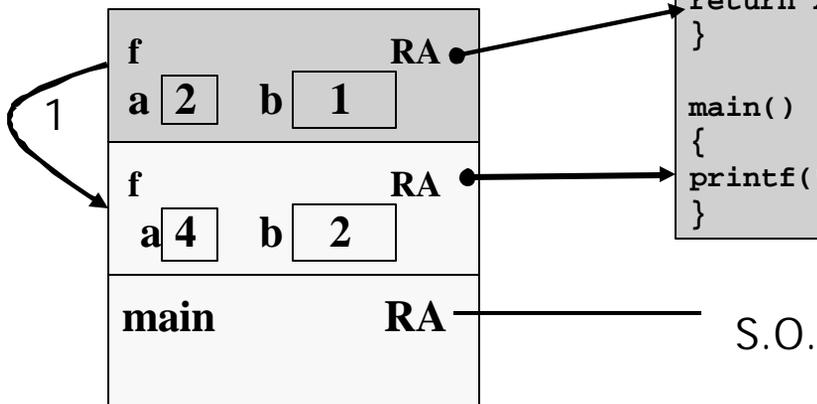
X [0]

Stack

Codice

```
int f(int a, int b)
{static int X=0;
if (!(a||b))
    return 1;
else
return X=f(a-2,b-1) + X;
}

main()
{
printf("%d\n", f(4,2));
}
```



Data Segment:

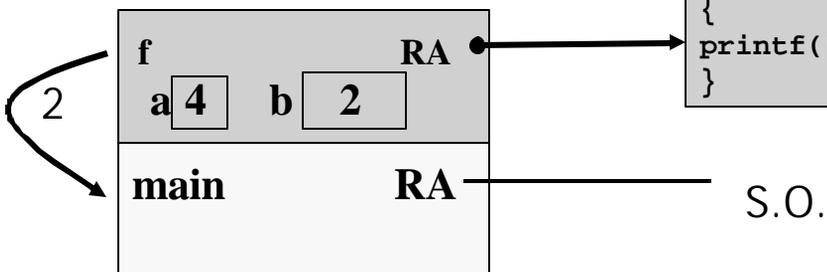
X 1

Stack

Codice

```
int f(int a, int b)
{static int X=0;
  if (!(a||b))
    return 1;
  else
    return X=f(a-2,b-1) + X;
}

main()
{
  printf("%d\n", f(4,2));
}
```



Data Segment:

X 2

Stack

Codice

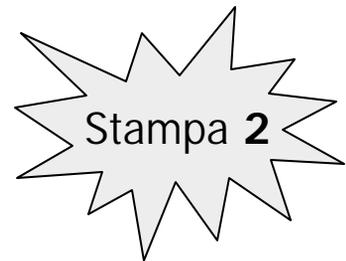
```
int f(int a, int b)
{static int X=0;
if (!(a||b))
    return 1;
else
return X=f(a-2,b-1) + X;
}

main()
{
printf("%d\n", f(4,2));
}
```

main

RA

S.O.



Data Segment:

X 2

Valori stampati:

Quindi, il programma stampa:

2

3. Esercizio “tipo” per prova scritta: analisi

Testo:

Dato il programma P (v. pagina successiva),
rispondere ai seguenti quesiti:

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

Codice del programma P da analizzare:

```
#include <stdio.h>
#define DIM 6
int f(int *a, int b)
{ if (*(a+b)!=0) return *(a+b);
  else return f(a,b+1)+b;
}
main()
{ int A[DIM]={0,0,0,0,0,0};
  int i;
  for(i=0; i<DIM; i+=3)
    A[i]=i;
  printf("%d\n", f(A,0));
  for(i=0; i<DIM; i++)
    printf("%d\t",A[i]);
}
```

Analisi

- Partendo dal main, il primo for esegue i seguenti assegnamenti:

$$A[0]=0$$

$$A[3]=3$$

- Al termine del ciclo, il vettore A e` quindi:

A

0	0	0	3	0	0
---	---	---	---	---	---

- Successivamente viene chiamata f...

Stack

Codice

```
int f(int *a, int b)
{...
  return f(a,b+1)+b;
}

main()
{...
  printf("%d\n",f(A,0));
  ...
}
```

main **RA**

A 0 0 0 3 0 0

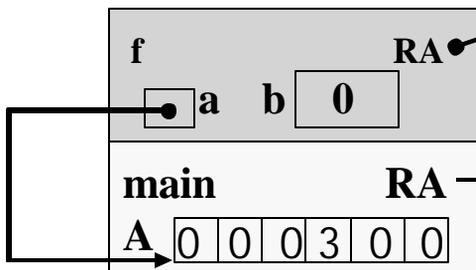
S.O.

Stack

Codice

```
int f(int *a, int b)
{...
  return f(a,b+1)+b;
}

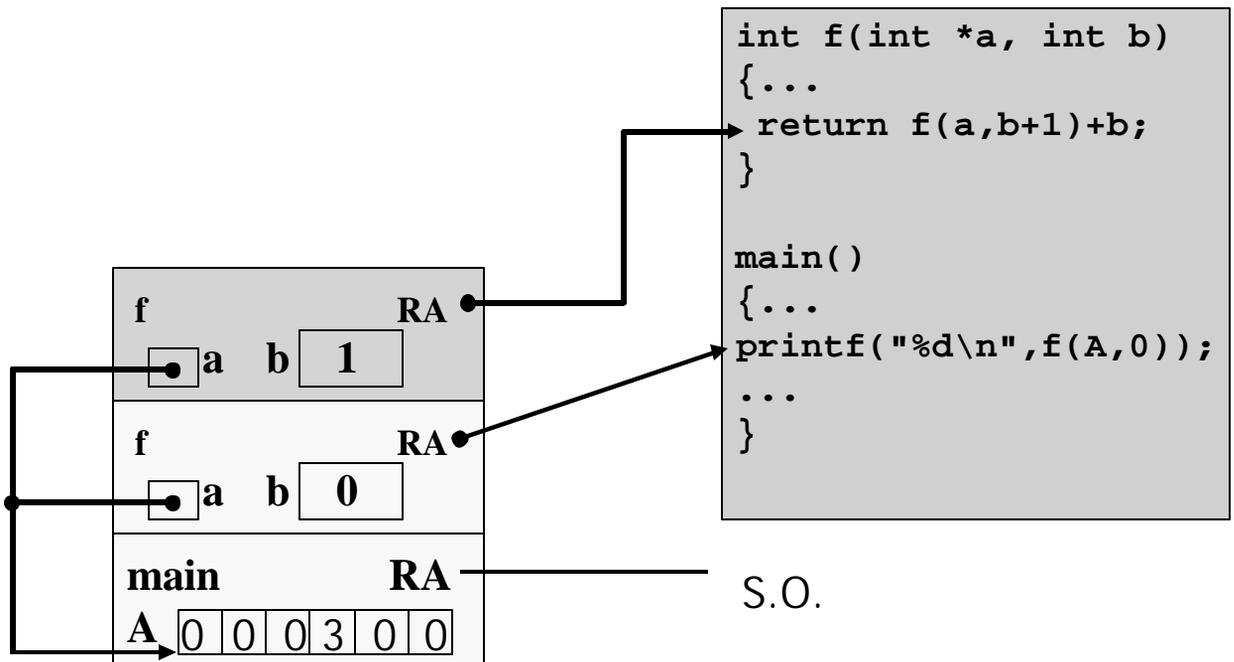
main()
{...
  printf("%d\n",f(A,0));
  ...
}
```



S.O.

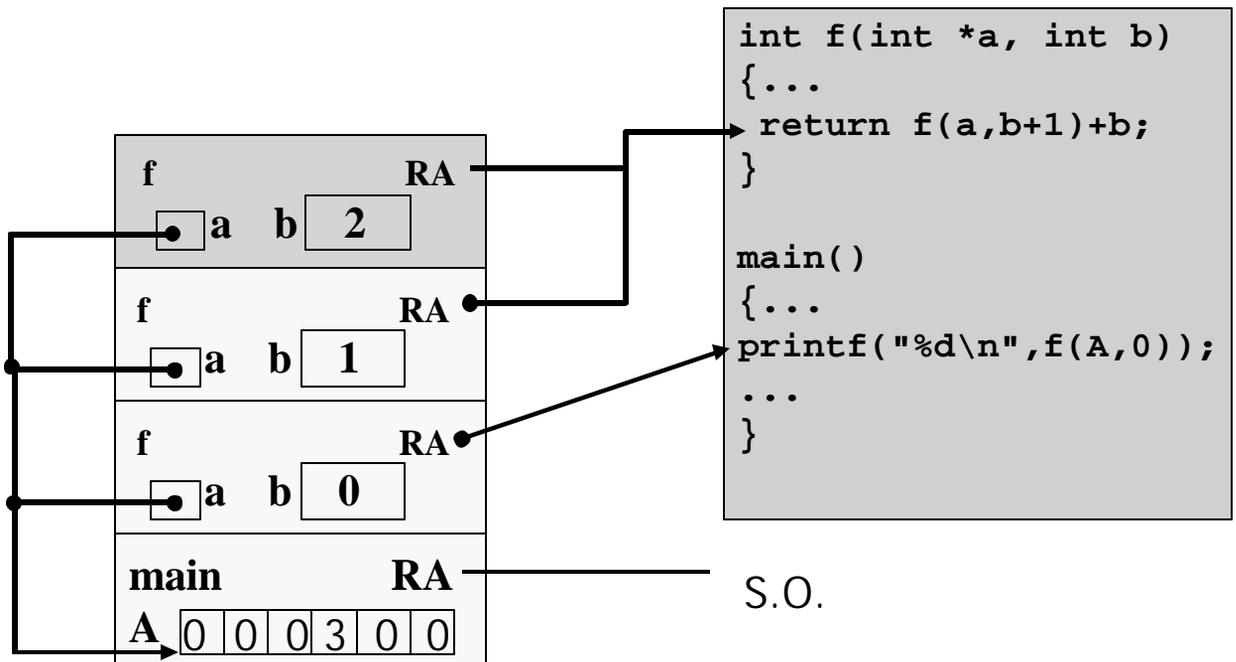
Stack

Codice



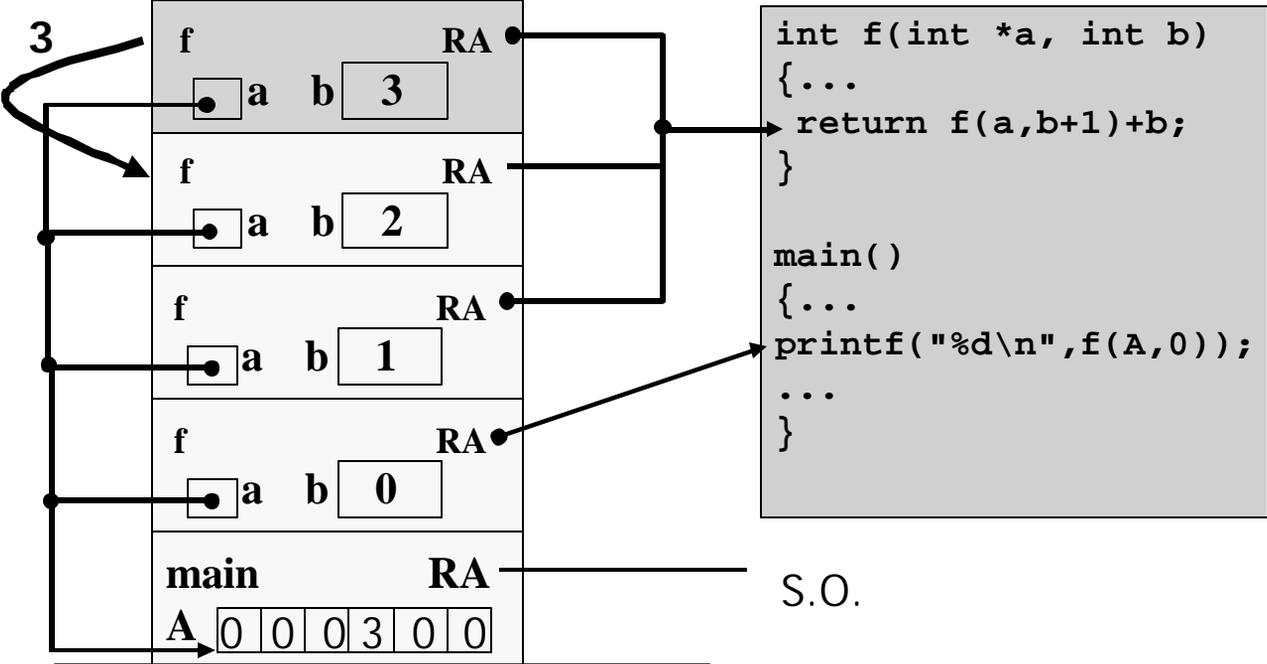
Stack

Codice



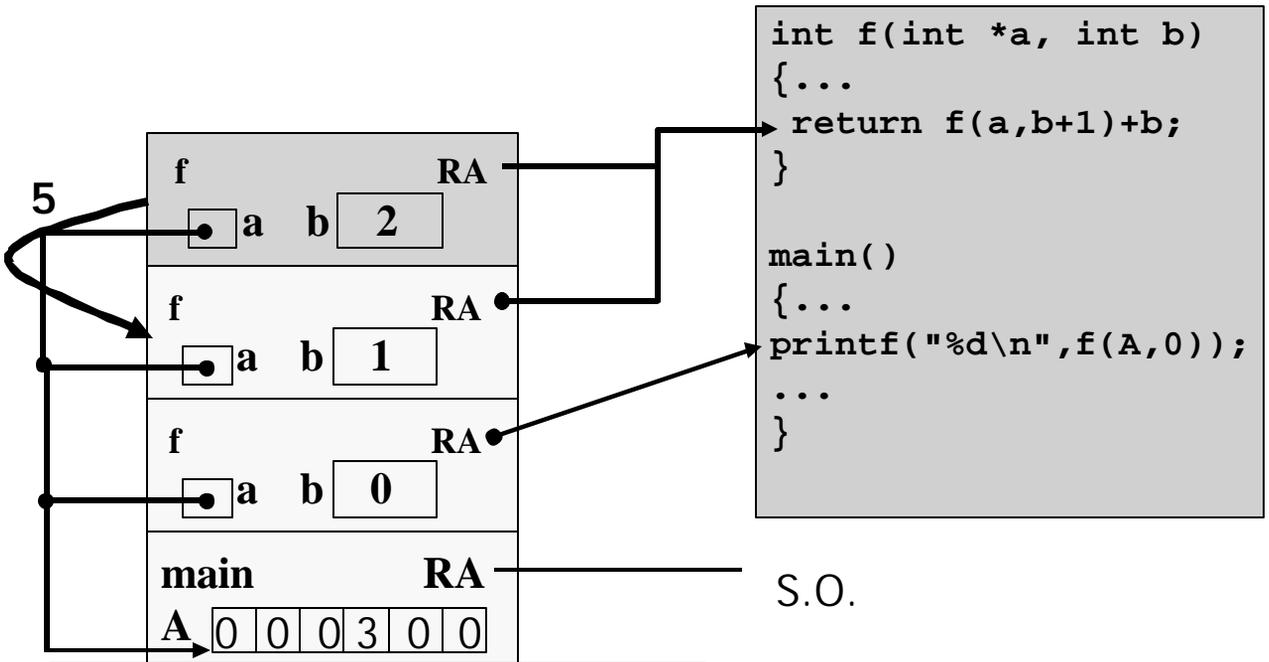
Stack

Codice



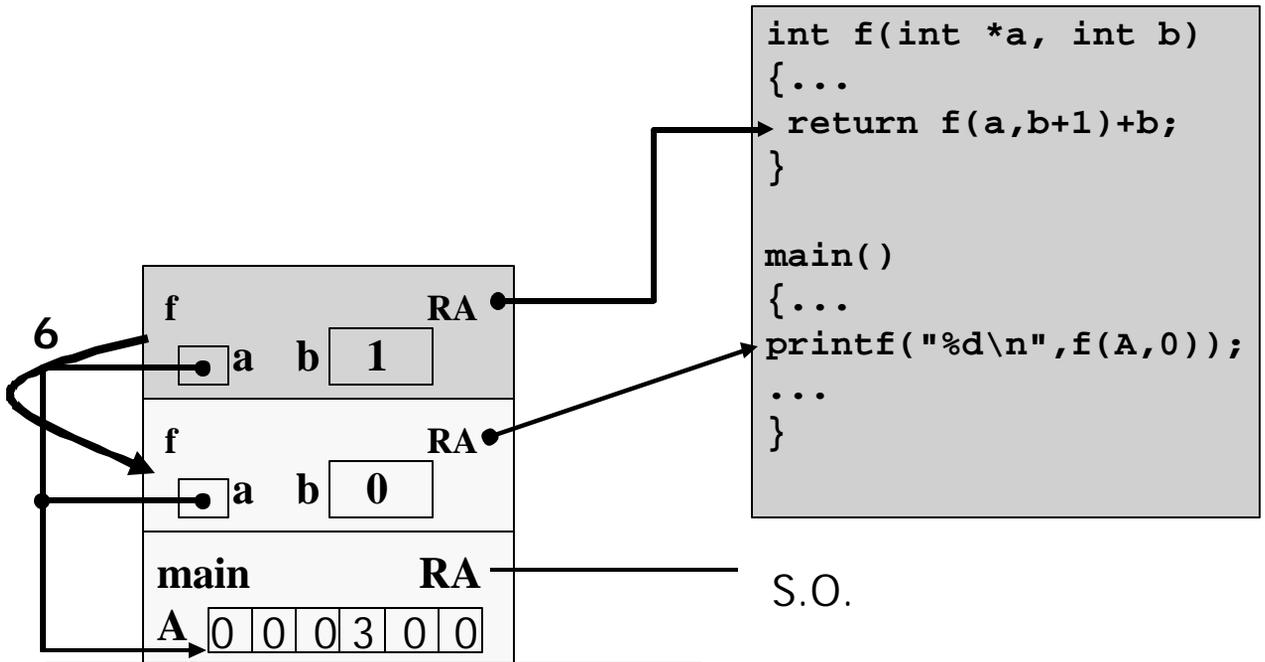
Stack

Codice



Stack

Codice

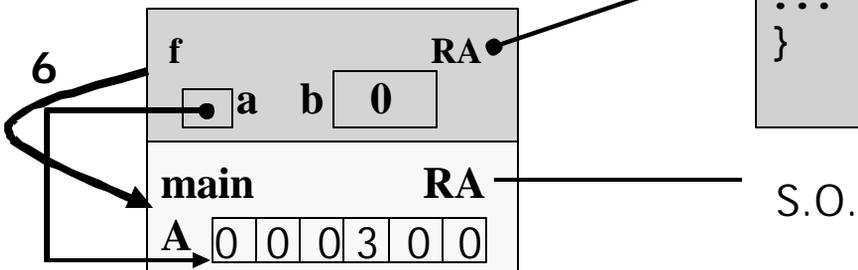


Stack

Codice

```
int f(int *a, int b)
{...
  return f(a,b+1)+b;
}

main()
{...
  printf("%d\n",f(A,0));
  ...
}
```



Stack

Codice

```
int f(int *a, int b)
{...
  return f(a,b+1)+b;
}

main()
{...
  printf("%d\n",f(A,0));
  ...
}
```

main	RA
A	0 0 0 3 0 0

S.O.

Stampa 6

Analisi

- Dopo la chiamata di f, viene quindi stampato il valore 6
- Successivamente, mediante il for, vengono stampati tutti gli elementi di A, e cioè:

A

0	0	0	3	0	0
---	---	---	---	---	---

Valori stampati:

Quindi, il programma stampa:

6

0 0 0 3 0 0