

Fondamenti di Informatica L-A (A.A. 2002/2003) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Mercoledì 25/06/2003 – durata 2h:30m

ESERCIZIO 1 (12 punti)

Si scriva una procedura **save()** che si occupi di popolare un file di testo chiamato **borsa.txt (2 punti)**. **save()** ha come parametro di ingresso un puntatore a file e deve richiedere all'utente l'inserimento interattivo da tastiera di dati da salvare sul file (puntatore passato come parametro). Il file di testo dovrà contenere richieste di operazioni di acquisto/vendita di azioni. Si preveda di terminare l'inserimento dati digitando, ad esempio, '0' come tipo di operazione (vedi seguito). Più precisamente, ogni sua riga include, nell'ordine:

- tipo operazione (carattere uguale a 'a' per l'acquisto, o a 'v' per la vendita), uno e un solo spazio di separazione;
- codice identificativo del cliente (numero intero), uno e un solo spazio di separazione;
- descrizione delle azioni (non più di 20 caratteri senza spazi), uno e un solo spazio di separazione;
- numero di azioni da acquistare/vendere (numero intero sempre positivo).

Ad esempio, **borsa.txt**:

```
a 1772 fiat 125
v 7453 hera 77
v 2437 olivetti 32
a 2332 hera 8
v 9825 microsoft 197
.....
```

Si scriva una procedura **load()** che riceva come parametro di ingresso un puntatore a file di testo e un carattere ('a' per gli acquisti, 'v' per le vendite) e restituisca come parametri di uscita un vettore **y** contenente strutture **operazione** (codice cliente, descrizione azione, numero azioni) e il numero degli elementi **N** inseriti in **y**. **load()** deve caricare in **y** le operazioni di acquisto se il parametro carattere vale 'a', le operazioni di vendita altrimenti (**4 punti**).

Si scriva un programma C che, utilizzando le procedure **save()** e **load()** precedentemente definite, inserisca in un vettore **z** (supposto di dimensione massima **DIM=100**) le sole operazioni di vendita contenute nel file **borsa.txt**. Il programma deve inoltre stampare a terminale tutti gli elementi di **z** relativi ad operazioni di clienti che hanno effettuato **esattamente due operazioni (6 punti)**.

ESERCIZIO 2 (5 punti)

Si scriva una funzione **f1()** che data in ingresso una lista di interi ed un numero intero **x**, restituisca in uscita una nuova lista, ottenuta dalla lista di partenza eliminando gli elementi che seguono, a distanza 2, eventuali occorrenze di **x**. Ad esempio, se invocata con **l=[11,7,8,0,7,-2,15,7]** e **x=7**, la funzione **f1()** deve restituire la lista **[11,7,8,7,-2,7]**.

La funzione **f1()** può essere realizzata in modo ricorsivo o iterativo, facendo riferimento al tipo di dato astratto **list** e alle sue operazioni primitive definite durante il corso (che quindi possono NON essere riportate nella soluzione), oppure utilizzando direttamente la rappresentazione collegata a puntatori.

ESERCIZIO 3 (7 punti)

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 L 20
void Proc(char [], int);
```

```

int N=L;

main () {
char *s;
int i;

s = (char *) malloc(L);
s = "Fondamenti di Infor";
Proc(s,L);
N--; N+2;
printf("N vale adesso: %d\n",N);
{
int i=0;
for (i=L-1; i>0; --i) printf("%c", *(s+i));
}
}

void Proc(char y[], int DIM) {
int i, N;
N++; i=1;
while (i<DIM/3) { y[i]=y[i+1]; i++; }
}

```

Esercizio 4 (5 punti)

Si consideri la seguente funzione F:

```

double F(int x){
    if (x<0) {x++; return 3*F(x/2);}
    else return -1;
}

```

Si scriva il risultato della funzione quando invocata come **F(-7)** e si disegnino i corrispondenti record di attivazione.

Esercizio 5 (2 punti)

Si consideri la grammatica G con scopo S e simboli terminali {**abs, exp, (,), x, y, z**}

```

S ::= A C D | D B | A
D ::= A B D | A B
C ::= A C C | B C
A ::= abs | exp
B ::= ( | ) | x | y | z

```

Si dica se la stringa **exp (abs y)** è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione *left most*.

Esercizio 6 (2 punti)

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:

96 - 23

SOLUZIONE

ESERCIZIO 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DIM 100

typedef struct {int cliente; char desc[21]; int numAz;} operazione;

void save(FILE * F) {
    char tipoOp, desc[21]; int cliente, numAz;
    do {
        printf("Inserire dati da salvare sul file. Per ogni operazione, tipo, codice
        cliente, descrizione azioni, numero azioni.\n Inserire 0 per terminare");
        scanf("%c %d %s %d\n", &tipoOp, &cliente, desc, &numAz);
        if (tipoOp!='0')
            fprintf(F, "%c %d %s %d\n", tipoOp, cliente, desc, numAz);
    }
    while (tipoOp!='0');
}

void load(FILE * F, char op, operazione y[], int *num) {
    char tipoOp, desc[21]; int cliente, numAz;
    *num=0;
    while (fscanf(F,"%c %d %s %d\n", &tipoOp, &cliente, desc, &numAz) != EOF)
    { if (op==tipoOp)
        { y[*num].cliente=cliente;
          strcpy(desc, y[*num].desc);
          y[*num].numAz=numAz;
          *num = *num+1; } }
}

main() {
    operazione z[DIM]; int N,i,j,count; FILE* f;

    if ((f=fopen("c:\\borsa.txt", "w"))==NULL) {
        printf("Non sono riuscito ad aprire il file in scrittura!"); exit(1); }
    save(f);
    fclose(f); f=fopen("c:\\borsa.txt", "r");
    load(f, 'v', z, &N);
    for (i=0; i<N; i++)
        { count=0;
          for (j=i; j<N; j++)
              if (z[i].cliente==z[j].cliente) count++;
          if (count==2)
              for (j=i; j<N; j++)
                  if (z[i].cliente==z[j].cliente)
                      printf("%d %s %d\n", z[j].cliente, z[j].desc, z[j].numAz);
        }
    fclose(f); }
```

ESERCIZIO 2

```
list fl(list l, int x) {
    list post;

    while (l!=NULL) && (l->next!=NULL) {
```

```

if (l->value==x)
    { post=l->next;
      if (post->next!=NULL) { post->next=(post->next)->next;
                            free(post->next); }
    }
l=post;
}

```

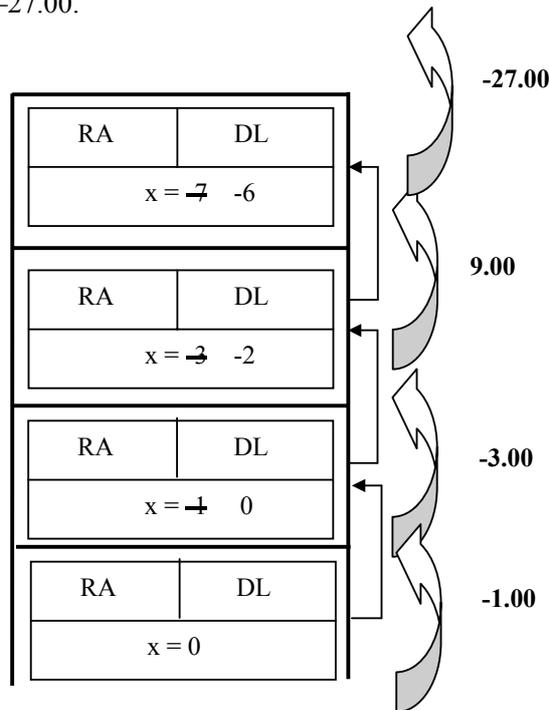
ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:
 N vale adesso: 19
 itneemadn (sulla stessa riga)

Infatti la stringa s viene inizializzata a “Fondamenti di Infor”, terminatore compreso. Successivamente viene invocata la procedura Proc(), di cui è stato dichiarato il prototipo prima del main(), alla quale si passa per riferimento la stringa (vettore di char). In particolare, Proc shifta a sinistra i caratteri della stringa di posizione da 1 a 6, estremi inclusi. Infine, il programma principale stampa a video alcuni caratteri della stringa s, nell’ordine da quello di posizione 9 a quello di posizione 1.

ESERCIZIO 4

La funzione restituisce il valore -27.00.



ESERCIZIO 5

La stringa è sintatticamente corretta. Derivazione *left most*:

S → D B → A B D B → exp B D B → exp (D B →
 → exp (A B B → exp (abs y)

ESERCIZIO 6

96-> 0 1100000
 -23-> 1 0010111

Tra i moduli dei numeri si esegue una sottrazione ottenendo: 0 1001001
 che vale 73 in base dieci.