

Fondamenti di Informatica L-A (A.A. 2005/2006) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame del 12/01/2006 - durata 2h30m
COMPITO A

ESERCIZIO 1 (13 punti)

Un negozio tiene traccia dei prodotti venduti registrandoli in un file di testo “**venduto.txt**”. Nella prima riga del file “**venduto.txt**” è presente un float e in ogni riga successiva, fino alla fine del file, la descrizione di un prodotto rappresentato nel seguente modo:

- due interi separati da spazi rappresentanti mese e giorno del mese (0 rappresenta il primo giorno del mese o il primo mese dell'anno),
- una stringa di al più 20 caratteri senza spazi rappresentante il nome del prodotto,
- un float rappresentante il prezzo del prodotto venduto.

Dopo aver creato opportunamente una struttura **prodotto** per contenere tali informazioni, il candidato realizzi:

1. una funzione

prodotto* leggi(FILE* fVen, int mese, int* len)

che, preso in ingresso un puntatore a file **fVen** e un intero **mese** rappresentante un mese (0=Gennaio,... 11=Dicembre), legga il file per sapere quanti prodotti sono stati venduti nel mese passato come parametro di ingresso o in mesi successivi a tale mese, sfrutti tale valore per allocare dinamicamente l'esatta memoria sufficiente a contenere tutti i prodotti del mese dato o di mesi successivi, e inserisca in tale area di memoria i soli prodotti che rispettano tutte le seguenti regole:

- a) il mese di vendita corrisponda a **mese** o ad uno successivo,
- b) il prezzo sia compreso nel range [**media-delta**, **media+delta**] dove **media** è il prezzo medio di tutti i prodotti presenti nel file **fVen** e **delta** il valore del float della prima riga del file **fVen**,
- c) non sia già stato inserito nell'area di memoria un prodotto con lo stesso nome.

La funzione **leggi(...)** restituisca alla funzione chiamante un puntatore all'area di memoria che contiene i prodotti che rispettano le regole date e, tramite **len**, il numero di prodotti inseriti nell'area di memoria.

2. una funzione **main()** che apra in modo opportuno il file “**venduto.txt**”, richieda all'utente il mese da cui bisogna iniziare per la selezione dei prodotti venduti, richiami opportunamente la funzione **leggi(...)**, scriva su un file binario di nome “**prodotti.bin**” i prodotti restituiti dalla funzione **leggi(...)**, ma solo se venduti tra il decimo e il ventesimo giorno (estremi inclusi).

È possibile utilizzare la funzione di libreria **strcmp(...)**. Si ricorda inoltre l'esistenza della funzione **void rewind(*FILE)** che riporta la testina di lettura ad inizio file.

ESERCIZIO 2 (9 punti)

Si scriva una funzione iterativa **compare(...)** che, ricevute in ingresso due liste di stringhe ben formate **l1** e **l2** (eventualmente anche di lunghezza diversa), restituisca una terza lista contenente gli elementi di **l1** di lunghezza maggiore o uguale ai corrispondenti elementi di **l2**. La lista restituita deve contenere anche tutti gli elementi di **l1** per cui non esiste un elemento corrispondente nella lista **l2**. Ad esempio, date due liste **l1= ["paperoga", "qui", "topolino", "minni", "gastone"]** e **l2=["pippo", "pluto", "etabeta"]**, la lista risultante deve essere **["paperoga", "topolino", "minni", "gastone"]** (l'ordine degli elementi della lista risultante non è rilevante).

A tal scopo si realizzi una funzione ricorsiva **length()** che, ricevuta in ingresso una stringa ben formata, restituisca la dimensione di tale stringa (carattere di terminazione non compreso).

La funzione **compare(...)** deve essere realizzata utilizzando il tipo di dato astratto **list**, definito per le stringhe (non è necessario riportare la definizione nella soluzione). Si possono utilizzare le sole operazioni primitive definite durante il corso, che quindi possono NON essere riportate nella soluzione. Non si possono usare altre funzioni di alto livello. In particolare la funzione **length(...)** deve essere realizzata senza l'utilizzo di funzioni di libreria; NON è possibile utilizzare la funzione **strlen(...)**.

ESERCIZIO 3 (6 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>
void removeC(char* stringa, char* caratteri, int* el){
    int trovato=0, w=0;
    char *c,*s=stringa;
    *el=0;
    while(*stringa!='\0'){
        trovato=0;
        c=caratteri;
        while((*c!='\0') && (trovato==0)){
            if(*stringa==*c) trovato=1;
            else c++;
        }
        if(trovato==0){
            s[w]=*stringa;
            w++;
        }
        else (*el)++;
        stringa++;
    }
    s[w]=*stringa;
}
int main(){
    char* s="ciao a tutti";
    char* del="a t"; // notare lo spazio tra 'a' e 't'
    int i;
    removeC(s,del,&i);
    printf("%s %d\n",s,i);
    return 0;
}
```

ESERCIZIO 4 (4 punti)

Data la funzione:

```
int fun(int a, int b){
    float f;
    f=a*1.5;
    if(f>b) return b;
    else{
        b=b-f;
        return 1+fun(a*2.1,b);
    }
}
```

e la funzione chiamante:

```
int main(){
    int i=fun(2,15);
    printf("%d\n",i);
    return 0;
}
```

mostrare la sequenza dei record di attivazione. Che cosa viene stampato sullo standard output?

Esercizio1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define DIM 21

typedef struct {
    int mese;
    int giorno;
    char nome[DIM];
    float prezzo;
} prodotto;

prodotto* leggi(FILE* venduti, int mese, int* len){
    int i, ok, tot=0, max=0; float somma=0, media, delta; prodotto el,*array;
    *len=0;
    fscanf(venduti,"%f",&delta);
    while( fscanf(venduti,"%d %d %s %f",
                &(el.mese), &(el.giorno), el.nome, &(el.prezzo))>0 ){
        if(el.mese>=mese) max++;
        somma=somma+el.prezzo; tot++;
    }
    media=somma/tot;
    array=(prodotto*)malloc(sizeof(prodotto)*max);
    if(array==NULL){
        printf("Errore con la malloc\n"); exit(-3);
    }
    rewind(venduti);
    fscanf(venduti,"%f",&delta);
    while( fscanf(venduti,"%d %d %s %f",
                &(el.mese), &(el.giorno), el.nome, &(el.prezzo))>0 ){
        if(el.mese>=mese){
            if(el.prezzo>=media-delta && el.prezzo<=media+delta){
                ok=1;
                for(i=0; i<*len && ok==1; i++){
                    if(strcmp(array[i].nome,el.nome)==0)ok=0;
                }
                if(ok){
                    array[*len]=el;
                    (*len)++;
                }
            }
        }
    }
    return array;
}

int main(){
    int mese,i,dim; FILE* f; prodotto* prodotti;
    if((f=fopen("venduto.txt","r"))==NULL){
        printf("Problemi durante l'apertura del file venduto.txt\n"); exit(-1);
    }
    printf("mese di interesse?\n");
    scanf("%d",&mese);
    prodotti=leggi(f,mese,&dim);
    fclose(f);

    if((f=fopen("prodotti.bin","wb"))==NULL){
        printf("Problemi durante l'apertura del file prodotti.bin\n"); exit(-2);
    }
    for(int i=0;i<dim;i++){
        if(prodotti[i].giorno>=9 && prodotti[i].giorno<=19){
            fwrite(&prodotti[i],sizeof(prodotto),1,f);
        }
    }
    fclose(f);
    return 0;
}
```

Esercizio2

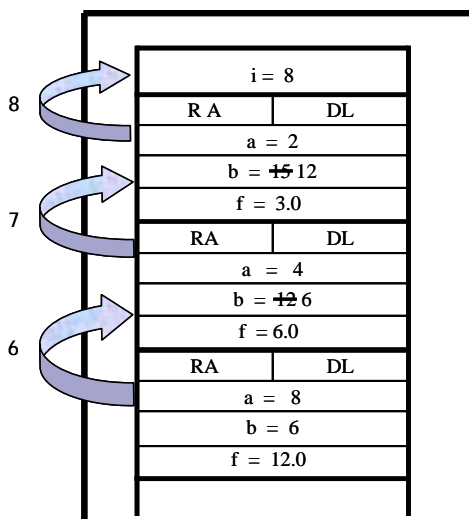
```
#include "list.h"
int length (char* s){
    if(*s=='\0') return 0;
    else{
        s++;
        return 1+length(s);
    }
}
list compare(list l1, list l2){
    char* s;
    int dimMin, dim;
    list lRet=emptylist();
    while(!empty(l1) && !empty(l2)){
        if(length (head(l1)) >= length(head(l2)) )
            lRet=cons(head(l1),lRet);
        l1=tail(l1);
        l2=tail(l2);
    }
    while(!empty(l1)){
        lRet=cons(head(l1),lRet);
        l1=tail(l1);
    }
    return lRet;
}
```

Esercizio3

La funzione `removeC(...)` prende in ingresso due stringhe ben formate ed un puntatore ad int. In particolare elimina dalla stringa `stringa` tutte le occorrenze dei caratteri presenti in `caratteri`. Il numero di caratteri eliminati viene inserito nella variabile puntata da `e1`.

Il main richiama la funzione `removeC(...)` richiedendo di eliminare dalla stringa `s` i caratteri presenti nella stringa `del` (caratteri `s`, `t` e `spazio`). Sullo standard output viene stampato:
`cioui 7`

Esercizio 4



Sullo standard output viene scritto il risultato della chiamata alla funzione `fun(...)`, ovvero
`8`