

**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 B**

**ESERCIZIO 1 (13 punti)**

Un comune tiene traccia delle multe effettuate ai cittadini in un particolare anno registrandole in un file di testo "multe.txt". Nella prima riga del file "multe.txt" è presente un int e in ogni riga successiva, fino alla fine del file, la descrizione di una multa rappresentata nel seguente modo:

- due interi separati da spazi rappresentanti mese e giorno del mese in cui è stata effettuata,
- due interi separati da spazi rappresentanti mese e giorno del mese in cui è stata pagata (giorno e mese hanno valore -1 se la multa non è stata ancora pagata, 0 rappresenta il primo giorno del mese o il primo mese dell'anno),
- una stringa di 7 caratteri senza spazi rappresentante la targa del veicolo multato,
- un float rappresentante l'ammontare della multa.

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

1. una funzione

```
multa* leggiMulta(FILE* fMulta, float delta, int* len)
```

che, preso in ingresso un puntatore a file **fMulta** ed un float **delta**, legga il file per sapere quante multe sono state pagate, sfrutti tale valore per allocare dinamicamente l'esatta memoria sufficiente a contenere tutte le multe pagate, e inserisca in tale area di memoria le sole multe che rispettano tutte le seguenti regole:

- a) la multa sia stata pagata,
- b) il mese in cui è stata fatta la multa corrisponda a **mese**, dove **mese** (0=Gennaio,... 11=Dicembre) è il valore dell'intero presente nella prima riga del file "multe.txt",
- c) l'ammontare non sia compreso nel range [**media-delta**, **media+delta**] dove **media** è l'ammontare medio di tutte le multe presenti nel file **fMulta**,
- d) non sia già stata inserita nell'area di memoria una multa relativa alla stessa targa.

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

2. una funzione **main()** che apra in modo opportuno il file "multe.txt", richieda all'utente il valore di **delta**, richiami opportunamente la funzione **leggiMulta (...)**, scriva su un file binario di nome "pagate.bin" i prodotti restituiti dalla funzione **leggiMulta (...)**, ma solo se pagate nei primi 10 giorni del mese (decimo compreso).

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 ricorsiva **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 minore 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=["legno", "fuoco", "aria", "ferro", "pioggia"]** e **l2=["acqua", "sabbia", "pietra"]**, la lista risultante deve essere [ "fuoco", "aria", "ferro", "pioggia"] (l'ordine degli elementi della lista risultante non è rilevante).

A tal scopo si realizzi una funzione iterativa **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)

```
void deleteC(char* parola, char* caratteri, int* num){
    int trovato=0, d=0;
    char *c,*s= parola;
    *num =0;
    while(*parola!='\0'){
        trovato=0;
        c=caratteri;
        while((*c!='\0') && (trovato==0)){
            if(*parola == *c) trovato=1;
            else c++;
        }
        if(trovato==1) (*num)++;
        else{
            *(s+d)=*parola;
            d++;
        }
        parola++;
    }
    *(s+d)=*parola;
}
int main(){
    char* s="happy new year";
    char* del="p e"; // notare lo spazio tra 'p' ed 'e'
    int i;
    deleteC(s,del,&i);
    printf("%s %d\n",s,i);
    return 0;
}
```

### **ESERCIZIO 4 (4 punti)**

Data la funzione:

```
int fun(int e, float x){
    float a;
    a=e*1.5;
    if(x>=a){
        x=x-a;
        return 1+fun(e*2,x);
    }
    else return x;
}
```

e la funzione chiamante:

```
int main(){
    int i=fun(2,14.3);
    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 8

typedef struct {
    int mese;
    int giorno;
    int mesePag;
    int giornoPag;
    char targa[DIM];
    float ammontare;
} multa;

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

int main(){
    int i,dim;          float delta;          FILE* f;          multa* multe;
    if((f=fopen("multe.txt","r"))==NULL){
        printf("Problemi durante l'apertura del file multe.txt\n"); exit(-1); }
    printf("delta di interesse?\n");
    scanf("%f",&delta);
    multe=leggiMulta(f,delta,&dim);
    fclose(f);
    if((f=fopen("pagate.bin","wb"))==NULL){
        printf("Problemi durante l'apertura del file pagare.bin\n"); exit(-2); }
    for(int i=0;i<dim;i++){
        printf("multe[%d].giornoPag %d\n",i,multe[i].giornoPag);
        if(multe[i].giornoPag<=9){
            fwrite(&multe[i],sizeof(multa),1,f);
        }
    }
    fclose(f);
    return 0;
}
```

## Esercizio2

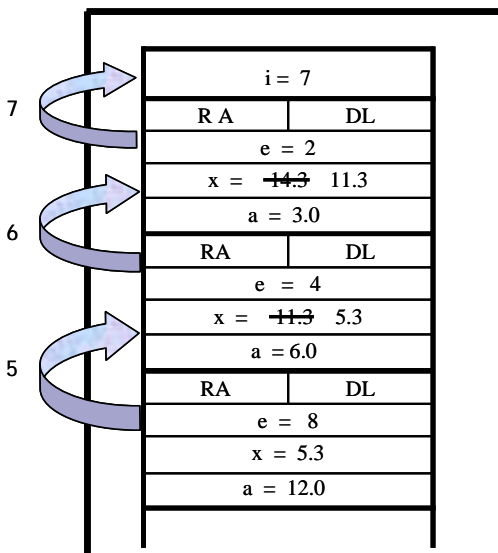
```
#include "list.h"
int length(char*s){
    int i=0;
    for(;*s!='\0';s++,i++);
    return i;
}
list compare (list l1, list l2){
    if(empty(l1) || empty(l2))
        return l1;
    else{
        if(length(head(l1)) < length(head(l2)))
            return cons( head(l1), compare(tail(l1),tail(l2)) );
        else return compare(tail(l1),tail(l2));
    }
}
```

## Esercizio3

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

Il main richiama la funzione `deleteC(...)` richiedendo di eliminare dalla stringa `s` i caratteri presenti nella stringa `del` (caratteri `p`, `e` e `spazio`). Sullo standard output viene stampato:  
`haynwyar 6`

## Esercizio 4



Sullo standard output viene scritto il risultato della chiamata alla funzione `fun(...)`, ovvero  
7