

Fondamenti di Informatica L-A (A.A. 2007/2008) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Giovedì 11 Settembre 2008 – durata 2h
Compito A

ESERCIZIO 1 (12 punti)

Un'autofficina vuole avvisare automaticamente i suoi clienti in caso sia scaduta la revisione periodica della loro macchina, ma solo se hanno sottoscritto uno specifico abbonamento. A tal fine l'autofficina mantiene l'elenco dei tagliandi fatti a tutti i clienti (abbonati e non); ciascun tagliando è rappresentato tramite una struttura dati che contiene la targa della macchina (un intero) e la data di scadenza del tagliando (un array di caratteri di dimensione sufficiente a contenere una stringa ben formata rappresentante la data nel formato aaaammgg, ad esempio "20080911" per l'11 Settembre 2008).

a) Il candidato realizzi una funzione:

```
int* check(tagliando* database, int* dim, char* oggi, int* abbonati, int dimAbbonati);
```

che riceva come parametri un vettore `database` di strutture tagliando (il candidato definisca opportunamente tale struttura), la dimensione logica di tale vettore (`dim`), una stringa ben formata `oggi` rappresentante la data odierna nel formato `aaaammgg`, un array di interi `abbonati` di dimensione `dimAbbonati` contenente le targhe delle macchine dei clienti abbonati.

La funzione deve restituire in uno spazio di memoria, allocato dinamicamente dalla funzione stessa non necessariamente della dimensione minima possibile, l'elenco delle targhe dei veicoli il cui tagliando sia già scaduto (quindi la data di scadenza è precedente alla data presente in `oggi`) e il cui cliente abbia sottoscritto un abbonamento. Inoltre la funzione `check()` deve restituire tramite `dim` la dimensione logica dell'array restituito.

b) Il candidato realizzi una semplice funzione `main()` di esempio che invochi correttamente la funzione `check()` e che stampi sullo standard output l'elenco delle targhe dei veicoli restituiti. A tal fine il candidato definisca due opportuni array per contenere l'elenco dei veicoli abbonati e il database dei tagliandi già effettuati. L'elenco degli abbonati viene popolato dal `main()` mentre il database viene popolato invocando la funzione `tagliando* leggiFile(char* filename, int* dim)` che il candidato NON deve realizzare.

Si ricorda l'esistenza della funzione `int strcmp(char* s1, char* s2)` per il confronto tra stringhe che restituisce un intero minore, uguale o maggiore di 0 a seconda che `s1` sia rispettivamente minore, uguale o maggiore di `s2`.

ESERCIZIO 2 (9 punti)

Si supponga di avere a disposizione, già definito, l'ADT lista per stringhe ben formate (denominato **list**, con relative primitive). Il candidato definisca una funzione ricorsiva

```
int conta(list l, char* car);
```

che, ricevute in ingresso una lista `l` ed una stringa ben formata `car`, restituisca la sommatoria totale del numero di occorrenze di tutti i caratteri presenti in `car` nella lista `l`.

Ad esempio, se `l = {"ciao", "mare", "montagna", "acqua"}` e `car = "caio"`, la funzione restituirà il valore 11 (4 per la prima stringa della lista `l`, 1 per la seconda, 3 per la terza e 3 per la quarta: quindi in totale 11).

ESERCIZIO 3 (5 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>

int load(int* to, int* from, int* dim){
    int i;
    *dim=0;
    for(i=0; i<10; i++){
        if(i%2==1){
            to[*dim]=from[i];
            (*dim)++;
        }
    }
    *(to+*dim)=-1;
    return -1;
}

int main(){
    int *a1, i, j;
    int a2[]={5,6,3,4,3,5,2,1,5,7};
    i=0;
    a1=(int*)malloc(sizeof(int)*((a2+3)+2));
    j=load(a1,a2,&i);
    printf("%d\n",i);
    while(*a1!=j){
        printf("%d ",a1[0]);
        a1++;
    }
    return 0;
}
```

ESERCIZIO 4 (3 punti)

Si consideri la grammatica G con scopo S, simboli non terminali {A, B, C, X, Y, Z} e simboli terminali {f, g, h, k, l, m, 2, 3, 4}:

```
S ::= AB | CA
A ::= YA | Y
B ::= ZC | YB
C ::= XC | ZY
X ::= k | l | m
Y ::= 2 | 3 | 4
Z ::= f | g | h
```

La stringa “342hkmg3” appartiene al linguaggio generato da tale grammatica?
In caso affermativo, se ne mostri la derivazione left-most.

ESERCIZIO 5 (3 punti)

Un elaboratore rappresenta i numeri interi su 8 bit tramite la notazione in complemento a 2. Indicare come viene svolta la seguente operazione aritmetica calcolandone il risultato secondo la rappresentazione binaria in complemento a 2 (si trasli anche il risultato in decimale per verificare la correttezza dell'operazione):

$$118 + (-43)$$

ESERCIZIO 1

```
typedef struct{
    char scadenza[9];
    int targa;
}tagliando;

int* check(tagliando* database, int*dim, char* oggi, int* abbonati, int
dimAbbonati){
    int i,j,temp=0;
    int* res=(int*)malloc(*dim*sizeof(int));
    for(i=0;i<*dim;i++){
        if(strcmp(database[i].scadenza,oggi)<0){
            int trovato=0;
            for(j=0; j<dimAbbonati&&!trovato; j++){
                if(abbonati[j]==database[i].targa){
                    res[temp]=database[i].targa;
                    temp++;
                    trovato=1;
                }
            }
        }
    }
    *dim=temp;
    return res;
}

int main(){
    tagliando* db;
    int dim,i;
    int* avvisi;
    int abb[]={111111,333333,555555,777777};
    db=leggiFile("database.txt",&dim);
    avvisi=check(db,&dim,"20080911",abb,4);
    for(i=0;i<dim;i++){
        printf("%d %d\n",i,avvisi[i]);
    }
    return 0;
}
```

ESERCIZIO 2

```
int conta(list l, char* car){
    if(empty(l)) return 0;
    else{
        int res=0;
        char*el=head(l);
        while(*el!='\0'){
            char*temp=car;
            while(*temp!='\0'){
                if(*el==*temp) res++;
                temp++;
            }
            el++;
        }
        return res+conta(tail(l),car);
    }
}
```

ESERCIZIO 3

Il programma è corretto sintatticamente e la sua esecuzione produce la stampa:

5

6 4 5 1 7

La funzione `main()` alloca spazio sufficiente per 6 interi ed in seguito invoca la funzione `load()`.

La funzione `load()` itera lungo i valori dell'array `a2` ed inserisce nello spazio di memoria allocato dal `main()` il valore presente nelle celle di indice dispari. Infine la funzione `load()` inserisce il valore `'-1'` come ultimo elemento in tale spazio di memoria e restituisce il valore `'-1'` alla funzione chiamante.

La funzione `main()` stampa sullo standard output in una riga il numero di elementi di `a2` copiati in `a1` ed in un'altra riga il valore di tali elementi (da notare che il valore `'-1'` viene utilizzato come elemento terminatore dell'array di interi).

ESERCIZIO 4

S -> AB -> YAB -> 3AB -> 3YAB -> 34AB -> 34YB -> 342B -> 342ZC -> 342hC -> 342hXC
-> 342hkC -> 342hkXC -> 342hkmc -> 342hkmZY -> 342hkmgY -> 342hkmg3

ESERCIZIO 5

118	->	01110110	01110110 + (118)
+43	->	00101011	11010101 = (-43)
		11010100	-----
-43	->	11010101	01001011 (+75)