

Fondamenti di Informatica T-1 (A.A. 2014/2015) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Martedì 13 Gennaio 2015 – durata 1h
Totale 12 punti, sufficienza con 7

Compito A

ESERCIZIO 1 (6 punti)

Date due liste di interi $l1$ e $l2$ e un intero $minVolte$, si realizzi una funzione RICORSIVA

```
list seleziona(list l1, list l2, int minVolte);
```

che restituisca una lista contenente gli elementi di $l1$ che compaiono almeno $minVolte$ volte nella lista $l2$. Per esempio, se $l1 = \{5, 4, 7\}$, $l2 = \{4, 2, 7, 4, 7, 7\}$ e $minVolte = 3$, la funzione `seleziona()` deve restituire la lista $l3 = \{7\}$.

A tal fine, si definisca e si utilizzi la funzione ITERATIVA

```
void conta(int e1, list l, int *count);
```

che dati in ingresso un intero $e1$ e una lista l restituisca il numero di ricorrenze di $e1$ nella lista l attraverso la variabile intera $count$ passata per riferimento.

Le funzioni `seleziona()` e `conta()` dovranno essere implementata utilizzando le primitive dell'ADT lista. Si realizzi inoltre una semplice funzione `main()` di prova che invochi correttamente la funzione `seleziona()` creata.

ESERCIZIO 2 (2 punti)

Si consideri la seguente grammatica G con scopo S , simboli non terminali $\{F, M, D, P\}$ e simboli terminali $\{x, y, 2, 4\}$

$S ::= MF \mid PMP$

$F ::= FM \mid FD \mid D$

$M ::= PM \mid P$

$D ::= x \mid y$

$P ::= 2 \mid 4$

La stringa "42y24" appartiene al linguaggio di tale grammatica?

In caso affermativo se ne mostri la derivazione left-most.

ESERCIZIO 3 (3 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 check(char *a){

    int l,i,OK=1;
    for(l=0;*(a+l)!='\0';) l++;
    l--;
    for(i=1;i>(l/2) && OK;i--){
        if(a[i]!=*(a+(l-i)))
            OK=0;
    }
    return OK;
}

int main(){

    char* s1="abcdcba";
    char* s2="ciao";
    int res1,res2;

    res1 = check(s1);
    res2 = check(s2);

    printf("%d\n%d\n", res1, res2);
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Si descriva rapidamente il concetto dei tipi di dato astratto stack e coda, discutendone le differenze.

Soluzioni

ESERCIZIO 1

```
void conta(int el, list l, int *count){
    while(!empty(l)){
        if(el == head(l))
            (*count)++;
        l=tail(l);
    }
}

list seleziona(list l1, list l2,int minVolte){

    int c = 0;

    if(empty(l1)||empty(l2))
        return emptyList();
    else{
        conta(head(l1),l2,&c);
        if(c>=minValue)
            return cons(head(l1),seleziona(tail(l1),l2,minVolte));
        else
            return seleziona(tail(l1),l2,minVolte);
    }
}

int main(){

    list a,b,c;

    a=cons(4,cons(2,cons(7,cons(4,cons(7,cons(7,emptyList())))));
    b=cons(5,cons(4,cons(7,emptyList())));

    c=seleziona(b,a,3);

    while(!empty(c)){
        printf("%d\n",head(c));
        c=tail(c);
    }
    return 0;
}
```

ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:
S → MF → PMF → 4MF → 4PF → 42F → 42FM (*) → 42FMM → 42DMM → 42yMM → 42yPM → 42y2M → 42y2P → 42y24

Derivazione alternativa, partendo da (*)
→ 42DM → 42yM → 42yPM → 42y2M → 42y2P → 42y24

ESERCIZIO 3

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

1
0

La funzione `main()` inizializza due puntatori a carattere `s1` e `s2` e dichiara due variabili intere `res1` e `res2`. Poi invoca due volte la funzione `check()`: la prima passando come argomento la stringa `s1` e assegnando il risultato alla variabile `res1`, poi passando come argomento la stringa `s2` e assegnando il risultato alla variabile `res2`.

La funzione `check()` inizializza a 1 una variabile intera `OK` e dichiara altre due variabili intere `l` e `i`. Esegue un primo ciclo `for` usando la variabile `l` come indice e iterando nell'area di memoria indicizzata partendo dal puntatore a carattere `a`, argomento della funzione, alla ricerca del terminatore di stringa, di fatto misurando la lunghezza della stringa allocata in quell'area di memoria. Decrementa di 1 il valore di `l` e, successivamente, esegue un secondo ciclo `for`. Questo ciclo usa la variabile `i` come indice, la inizializza al nuovo valore di `l` e la decrementa di 1 ad ogni ciclo finché non raggiunge la metà della stringa (`l/2`) o interpreta la variabile `OK` come falsa. Ad ogni ciclo, verifica se il carattere della stringa `a` in posizione `i` sia diversa da quella in posizione `l-i`. Concettualmente, sono due posizioni equidistanti dal centro della stringa, quindi `check()` verifica se la stringa è palindroma. Infatti, qualora il controllo dentro al secondo `for` fallisca, alla variabile intera `OK` viene assegnato il valore 0, cioè falso. Terminato il ciclo, la funzione `check()` restituisce il valore contenuto nella variabile `OK`.

La funzione `main()` stampa sullo standard output i due valori interi delle variabili `res1` e `res2` (1 e 0).