Fondamenti di Informatica T-1 (A.A. 2009/2010) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Mercoledì 13 Gennaio 2010 – durata 1h
Totale 12 punti, sufficienza con 7
Compito B

ESERCIZIO 1 (6 punti)

Si supponga di avere a disposizione l'ADT lista definito per le stringhe ben formate.

Il candidato definisca la funzione *ricorsiva*

```
int parole(list stringList, char* substring);
```

che restituisce il numero di elementi della lista **`stringList`** che NON iniziano con i caratteri della stringa ben formata **`substring`**. Ad esempio, se **`stringList = { "casa", "muro", "maestro", "montagna" }`**, e **`substring = "ma"`**, la funzione **`parole(...)`** restituisce 3 dato che **`"casa"`**, **`"muro"`** e **`"montagna"`** NON iniziano per **`"ma"`**. La funzione **`parole`** deve essere implementata utilizzando le sole primitive dell'ADT lista.

A tal scopo si realizzi e si utilizzi la funzione di supporto

```
int nonIniziaPer(char* string, char *subs);
```

che, date due stringhe ben formate **`string`** e **`subs`**, restituisce 1 se la stringa **`string`** NON inizia esattamente coi caratteri della stringa **`subs`**, 0 altrimenti. Per ipotesi la stringa ben formata **`subs`** non è mai più lunga della stringa ben formata **`string`**.

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione:

```
int fun(int a, float b){
      int x;
      x=b*a;
      if( (x%9)==0)
            return 3;
      else
            return b - fun(a, x);
}
```

Mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametri attuali (3.7, 3.7).

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* create(int* a1, int* a2, int dim1, int *dim2){
    int* b, *temp;
    b=(int*)malloc(sizeof(int)*(dim1+*dim2));
    temp=b;
    while(*dim2>0 && (*a1)!=(*a2)){
        if( ((*dim2)%2)==1 ){
            *b=*a1;
            a1++;
        }
        else{
            *b=*a2;
            a2--;
            (*dim2)--;
        }
        b++;
    }
    *dim2=b-temp;
    return temp;
}

int main(){
    int *res, i;
    int dim2=3;
    int a[]={3,4,5,6,7,8,9};
    int b[]={6,7,8};
    res=create(a, b+dim2-1, 7, &dim2);
    for(i=dim2-1; i>=0; i--){
        printf("%d\n", res[dim2-i-1]);
    }
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Il seguente codice C compila correttamente?
Motivare opportunamente la risposta data.

```
int x[] = {4, 5, 6};
int y[] = {1, 2, 3};
y=x;
```

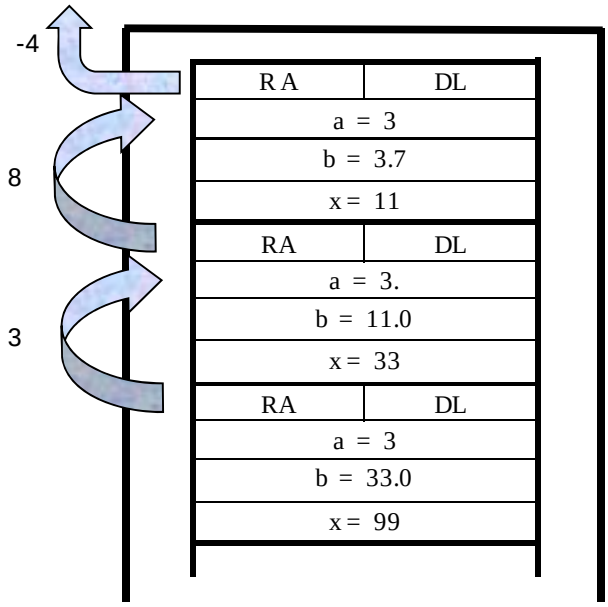
Soluzioni

ESERCIZIO 1

```
int nonIniziaPer(char* string, char *subs){
    int ok=0;
    // per ipotesi s sempre più lunga di subs
    while(ok==0 && (*subs!='\0') ){
        if(*string!=*subs){
            ok=1;
        }
        string++; subs++;
    }
    return ok;
}
int parole(list l, char* substring){
    char* s;
    if(empty(l)){
        return 0;
    }
    else{
        s=head(l);
        return nonIniziaPer(head(l), substring) +
                                              parole(tail(l),substring);
    }
}
```

ESERCIZIO 2

La funzione restituisce il valore -4.



ESERCIZIO 3

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

```
3
4
5
6
7
```

La funzione `main()` invoca la funzione `create()` con parametri attuali rispettivamente l'indirizzo della prima cella di memoria dell'array `a`, l'indirizzo dell'ultima cella di memoria dell'array `b`, l'intero 7 e l'indirizzo della variabile `dim2`.

La funzione `create()` alloca spazio sufficiente a contenere 21 interi. Sia la variabile `b` che la variabile `temp` della funzione `create()` puntano alla prima cella della memoria allocata dinamicamente. Notare che la variabile `b` della funzione `create()` non ha alcuna relazione con la variabile `b` della funzione `main()`.
La funzione procede ad assegnare alle celle di memoria appena allocate il valore delle celle degli array `a1` (se il valore referenziato da `dim2` è dispari) ed `a2` (se il valore referenziato da `dim2` è pari), scorrendo `a1` dall'inizio verso la fine ed `a2` dalla fine verso l'inizio. Infine la funzione `create()` assegna alla cella di memoria referenziata da `dim2` il valore corrispondente al numero di celle dell'array `b` correttamente inizializzate e restituisce al chiamante l'indirizzo della prima cella di memoria allocata dinamicamente.

La funzione `main()` infine stampa a video (dalla prima all'ultima) le celle di memoria correttamente inizializzate dalla funzione `create()`.