

**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 A**

**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 words(list stringList, char* substring);
```

che restituisce il numero di elementi della lista **stringList** che iniziano esattamente con i caratteri della stringa ben formata **substring**. Ad esempio, se **stringList** = { "casa", "mare", "montagna", "maestro" }, e **substring** = "ma", la funzione **words(...)** restituisce 2 dato che "mare" e "maestro" iniziano per "ma". La funzione **words** deve essere implementata utilizzando le sole primitive dell'ADT lista.

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

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

che, date due stringhe ben formate **string** e **subs**, restituisce 1 se la stringa **string** 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(float x, int y){
    int z;
    z=x*y;
    if( (y%3)==1)
        return 0;
    else
        return x + fun(x, z);
}
```

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)==0 ){
            *b=*a1;
            a1++;
        }
        else{
            *b=*a2;
            a2--;
            (*dim2)--;
        }
        b++;
    }
    *dim2=b-temp;
    return temp;
}

int main(){
    int *res, i;
    int dim2=3;
    int a[]={1,2,3,4,5,6,7};
    int b[]={4,5,6};
    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)

Illustrare sinteticamente la differenza fra approccio compilato, interpretato ed intermedio.

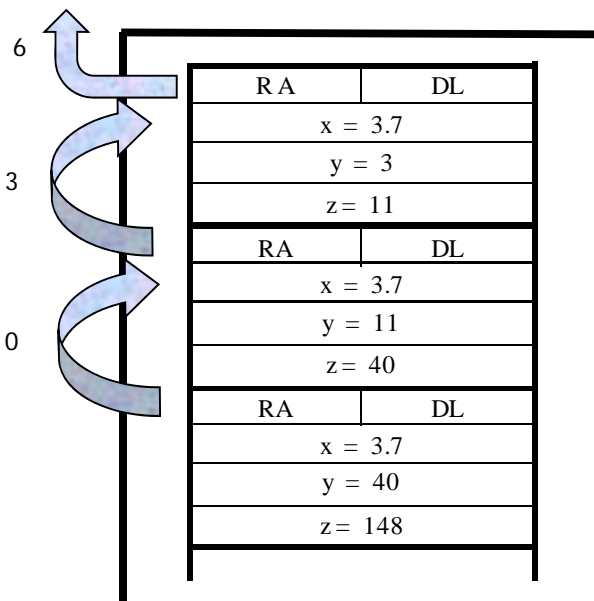
## Soluzioni

### ESERCIZIO 1

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

### ESERCIZIO 2

La funzione restituisce il valore 6.



### **ESERCIZIO 3**

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

6  
1  
2  
3  
4

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 pari) ed `a2` (se il valore referenziato da `dim2` è dispari), 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()`.