

**Fondamenti di Informatica T-1 (A.A. 2013/2014) - Ingegneria Informatica**  
**Prof.ssa Mello**  
**Prova Parziale d'Esame di Giovedì 11 Settembre 2014 – durata 1h**  
**Totale 12 punti, sufficienza con 7**

**ESERCIZIO 1 (6 punti)**

Dati due interi  $a$  e  $b$  e una lista di interi  $l$ , si realizzi una funzione ITERATIVA

```
char control(int a, int b, list l);
```

che restituisca il carattere 'y' nel caso in cui il prodotto tra  $a$  e  $b$  sia presente tra gli elementi della lista  $l$ , il carattere 'n' altrimenti. Per esempio, se  $a=5$ ,  $b=3$  e  $l=[78,5,15]$ , la funzione `control()` deve restituire il carattere 'y', in quanto  $3 \times 5 = 15$ .

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

```
int multiplication(int a, int b);
```

che dati in ingresso due interi  $a$  e  $b$ , ne calcoli il prodotto e ne restituisca il risultato.

La funzione `control()` dovrà essere implementata utilizzando le primitive dell'ADT lista; ogni altra funzione (compresa la funzione `multiplication()`) dovrà essere opportunamente specificata dal candidato. Si realizzi inoltre una semplice funzione `main()` di prova che invochi correttamente la funzione `control()` creata.

**ESERCIZIO 2 (2 punti)**

Si consideri la seguente grammatica  $G$  con scopo  $S$ , simboli non terminali  $\{A, B\}$  e simboli terminali  $\{b, d, p, q\}$

$S ::= A \mid B \mid SS$

$A ::= pSq$

$B ::= bd$

La stringa "**bdppbdqbdq**" 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>
#include <string.h>

void fun(int *d1, int *d2, int *ln, char *temp){
    int i = 0;

    while( *temp != 'l' ){
        temp++;
    }

    while(i < *ln){
        if(*d1 <= 24){
            *d2 = *d2 + *d1;
            ++d2;
        }
        else{
            temp = temp + 2;
            printf("%s\n", temp);
        }
        d1++;
        i++;
    }
}

int main(void){
    int i, ln = 2;
    int d1[4] = {2,34,3};
    int d2[4] = {3,8,12};
    char *s1 = "alice";

    fun(d1, d2, &ln, s1);

    for (i=0; i<ln; i++){
        printf("%d\n",d2[i]);
    }
    return (0);
}
```

### **ESERCIZIO 4 (1 punto)**

Il candidato illustri brevemente a cosa serve e cosa tipicamente contiene il file di intestazione (header file) e specifichi come è possibile includerlo nel corrispondente file con estensione '.c'.

# Soluzioni

## ESERCIZIO 1

```
int multiplication(int a, int b){
    if(b == 0){
        return 0;
    }
    else{
        return a + multiplication(a, b-1);
    }
}

char control(int a, int b, list l){
    int product = multiplication(a,b);
    while(!empty(l)){
        if(head(l) == product){
            return 'y';
        }
        l = tail(l);
    }
    return 'n';
}

int main(){
    list t = cons(78,cons(5,cons(15, emptylist())));
    char res = control(5,3,t);
    printf("Risposta: %c\n", res);
    return 0;
}
```

## ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:  
 $S \rightarrow SS \rightarrow BS \rightarrow bdS \rightarrow bdA \rightarrow bd pSq \rightarrow bd pSSq \rightarrow bd pASq \rightarrow bdppSqSq \rightarrow bdppBqSq \rightarrow bdppbdqSq$   
 $\rightarrow bdppbdqBq \rightarrow bdppbdqbdq$

## ESERCIZIO 3

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

```
ce
5
8
```

La funzione `main()` inizializza un intero `ln`, due array di interi `d1` e `d2` e un puntatore a carattere `s1`. Poi invoca la funzione `fun()`.

La funzione `fun()` sposta inizialmente il puntatore della variabile `temp` sulla cella che contiene il carattere `'1'`. Il secondo ciclo `while` itera fintanto che l'indice `i` (inizializzato a 0) risulta essere minore di `ln` (2) e controlla se il valore puntato da `d1` è  $\leq$  di 24. Se sì, assegna alla cella di memoria attualmente puntata da `d2` la somma dei valori attualmente puntati da `d2` e `d1`, altrimenti incrementa il puntatore `temp` di due posizioni e stampa sullo standard output ciò che rimane della stringa ("ce").

La funzione `main()` stampa sullo standard output i primi due valori interi contenuti in `d2` (5 e 8).