FONDAMENTI DI PROGRAMMAZIONE - CdL in MATEMATICA
PROVA SCRITTA DEL 11/1/2013

Scrivere **in stampatello** COGNOME, NOME e MATRICOLA su ogni foglio consegnato

N.B.: Negli esercizi di programmazione, viene valutata anche la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione (come, ad esempio, `continue`, `break` e istruzioni di `return` all'interno di cicli che ne provochino l'uscita forzata). Infine non è consentito l'uso di variabili statiche.
Laddove è utilizzato, il tipo `boolean` è definito da `typedef enum {false, true} boolean;`

ESERCIZIO 1 (6 punti)
Sia data la seguente grammatica libera sull'alfabeto $\Sigma = \{a, b, c\}$.

$$\begin{array}{lcl} S & ::= & aSb|aAb \\ A & ::= & aA|B \\ B & ::= & bB|b \end{array}$$

- Scrivere il linguaggio generato dalla grammatica.
- Dire se il linguaggio è libero o regolare.

SOLUZIONE ESERCIZIO 1

$$\begin{array}{lcl} L(S) & = & aaa^*bb^+b \cup aa^*bb^+ = \{a^n b^m | n > 0 \wedge m > 1\} \\ L(A) & = & a^*b^+ \\ L(B) & = & b^+ \end{array}$$

ESERCIZIO 2 (7 punti)
Scrivere una funzione *ricorsiva* che, dato `n`, stampi la somma di tutti gli interi dispari $m \leq n$ che dividono n.

SOLUZIONE ESERCIZIO 2

```
int Divide(int n,int m)
{
int i =0;
if (m != 0)
   {if (n \% m = 0)
        {if (m \%2 = 1) {i = m};
        return i + Divide(n,m-1);
}
```

ESERCIZIO 3 (7 punti)
Dato un numero naturale, espresso come costante N, si rappresenti un sottoinsieme S dei naturali da 1 a N come un array a_S di dimensione N di booleani tali che $a_S[i] = true$ se e solo se i appartiene ad S. Scrivere una funzione **iterativa** che dati tre array, di cui i primi due corrispondenti a due sottoinsiemi S_1, S_2, popoli il terzo array in modo da farlo corrispondere alla loro intersezione $S_1 \cap S_2$.

ESERCIZIO 4 (11 punti)
Date due liste di interi,

(i) definire gli opportuni tipi di dato per rappresentare il tipo di dato lista di interi;

(ii) progettare una procedura **iterativa** per combinare – senza allocare nuovi nodi – le due liste in modo che alternino elementi della prima e della seconda lista, senza alterarne l'ordine: [primo nodo della prima lista, primo nodo della seconda lista, secondo nodo della prima lista, secondo nodo della seconda lista, ...]. L'unico vincolo è che i primi due elementi non siano uguali. Se le due liste cominciano con lo stesso elemento il primo nodo della prima lista deve essere eliminato.

(iii) progettare una funzione **ricorsiva** che, presa in ingresso attraverso opportuni parametri la lista risultante dalla procedura sopra, e dato un numero n, calcoli il numero di occorrenze di n tra gli elementi della lista.