


**Fondamenti di Informatica - Ingegneria delle Telecomunicazioni – Prof. Maristella Matera**

Seconda prova in itinere – 9 Febbraio 2015

Il tempo massimo a disposizione per svolgere la prova è di **2h e 15m**. Non è permessa la consultazione di alcun materiale didattico ed è vietato utilizzare calcolatrici, telefoni, PC. **Il voto minimo per superare la prova è 18.**

**Esercizio 1 – Liste dinamiche (8 punti)**

Scrivere una funzione in C che data una lista dinamica la modifichi eliminando tutti gli elementi di posizione pari nella sequenza.

**Esercizio 2 – Funzioni in C e lettura da file (12 punti)**

Il quadrato di Polibio è una tecnica di cifratura del testo inventata dal celebre storico greco. La chiave per cifrare il testo è data da una matrice, **char key[5][5]**, che memorizza in un ordine casuale le 25 lettere dell'alfabeto (I e J usano la stessa codifica). Per ottenere la cifratura di una parola, si sostituisce ogni lettera con le relative coordinate nella matrice, giustapposte in modo da formare un numero intero di due cifre.

Sia per esempio data la matrice key sotto riportata, in cui per semplicità le lettere sono memorizzate in ordine alfabetico. In base a tale matrice, la cifratura della parola **mare** è: **<32, 11, 42, 15>**.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Infatti: **M** = key[3,2]; **A** = key[1,1]; **R** = key [4,2], **E** = key[1,5].

N.B.: Le coordinate usate nella cifratura partono da 1, mentre gli indici per scandire la matrice devono partire da 0!

Supponendo di dover cifrare un testo contenuto in un file, definire una funzione avente il seguente prototipo:

```
void polybius(FILE *fp, char key[][5], int *cifratura)
```

che, a partire da:

- il **file puntato da fp**, che contiene **un testo lungo massimo N caratteri (spazi inclusi)**, con N costante nota a priori e definita nella parte dichiarativa del programma;
- la **matrice quadrata key** di dimensione 5 x 5 che contiene la chiave di cifratura,

costruisca l'**array di interi, cifraura**, in modo che questo memorizzi, secondo lo schema sopra descritto, la cifratura del testo letto dal file. **Gli eventuali spazi presenti nel testo sono codificati nell'array con il valore 0. Si assuma inoltre che nel testo non siano inclusi segni di punteggiatura e caratteri diversi da quelli presenti nella matrice.**

**Esercizio 3 – Funzioni ricorsive (8 punti)**

Sia data una matrice quadrata di dimensione NxN, con N costante dichiarata nella parte dichiarativa globale del programma. Scrivere **una funzione ricorsiva in C** che restituisca il valore 1 se la matrice possiede almeno due righe consecutive uguali, 0 altrimenti.

**Esercizio 4 – Processi (6 + 1 punti)**

Considerando il seguente codice, contenuto nel file **processi.c** :

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv){
    int i=2, j=0, k=0;
    if (argc<=1) return -2;
    j=strlen(argv[1]);
    i=strlen(argv[0]);
    for (; i>2; i--){
        int pid=fork();
        if (!pid){
            i-=2;
            printf("%d\t%c\n",getpid(),argv[1][j-i]);
            i-=3;
        } else {
            printf("%d\t%d\n", getpid(), getppid());
            wait();
        }
    }
    return i;
}
```

Determinare l'output del programma e la struttura dei processi quando il programma viene compilato e invocato con le seguenti linee di comando (assumendo che il primo processo abbia pid=1000 e nessun processo esterno venga mandato in esecuzione nel sistema durante l'esecuzione del programma):

```
gcc ./processi.c
./a.out <matricola>
```

**Bonus (1 punto):** dire inoltre quale warning viene emesso dal compilatore e come modificare il codice per evitarlo.