



Prova Scritta del 15-02-2019

Esercizio 1.

Si implementi una funzione ricorsiva in C++ che riceva in input un array A di caratteri di dimensione N, un array B di interi di dimensione N-1, una matrice C di caratteri di dimensione L x M, scorrendo la matrice da destra a sinistra e dall'alto al basso, individui se ciascun carattere A[i] di A occorre in C ed il successivo carattere si trova a distanza B[i]. In tal caso, si stampino i caratteri della matrice che stanno tra i caratteri di A, altrimenti si stampi "ERRORE".

N.B Lo svolgimento corretto dell'esercizio senza l'utilizzo della ricorsione vale metà punteggio.

Esempio 1. Se abbiamo un array A = {c,a,s,a,l,e}, un array B = {6,2,3,2,1} e una matrice C come segue:

m	n	c	i	L	m
i	o	a	p	s	a
p	a	a	l	e	c
v	b	w	s	x	u

In questo caso si stamperà: **ilmiopapa**

Infatti

- il primo carattere di A, **c**, si trova in posizione [0][2];
- a distanza 6, troviamo il secondo carattere di A, **a**;
- a distanza 2, da quest'ultimo, troviamo il terzo carattere di A, **s**;
- a distanza 3, da quest'ultimo, troviamo il quarto carattere di A, **a**;
- a distanza 2, da quest'ultimo, troviamo il quinto carattere di A, **l**;
- a distanza 1, da quest'ultimo, troviamo il sesto carattere di A, **e**.

NOTA: Lo svolgimento corretto senza l'utilizzo della ricorsione vale metà punteggio.

Esercizio 2. (SOLO PER GLI STUDENTI DALL'A.A. 2015/2016 IN POI)

Si consideri la classe CANTANTE riportata di seguito, che si può supporre interamente implementata.

```
class Cantante {
friend ostream& operator>>(ostream&, Cantante&);
friend ostream& operator<<(ostream&, const Cantante&);
private:0
    string nome;
    unsigned eta;
    string genere;
    unsigned numeroDischi;
public:
    Cantante();
    Cantante(const string&, unsigned, const string&, unsigned);
    const string& getNome() const;
    unsigned getEta() const;
    const string& getGenere() const;
    unsigned getNumeroDischi() const;
```

**Prova Scritta del 15-02-2019**

};

In ciascuna serata del festival di Sanremo si esibiscono una serie di cantanti. Alcuni cantanti sono in gara, altri invece sono ospiti. Il sistema deve prevedere la possibilità che si aggiunga un nuovo cantante alla gara e che decida di invitare un nuovo cantante ospite in una delle serate. Il sistema, inoltre, deve consentire di conoscere il genere per il quale sono previsti più cantanti in gara. A tale scopo, si implementino tutti i metodi della classe SERATA, la cui interfaccia è riportata di seguito:

```
class Serata{
friend ostream& operator<<(ostream&, const Serata&);
private:
    Cantante* cantanti;
    bool* inGara;
    unsigned n;
public:
    Serata();
    Serata (Cantante*, bool*, unsigned);
    Serata (const Serata&);
    ~Serata();
    Serata& operator=(const Serata&);
    bool aggiungiCantante(const Cantante&, bool inGara);
    bool eliminaCantante(unsigned);
    unsigned getNumeroCantanti() const;
    string getGenerePiuPresente() const
};
```

Si implementi inoltre, un main che stampi alcune informazioni su un festival. In particolare, occorre leggere da input un array di 5 serate che compongono il festival e stampare:

- i cantanti, tra quelli in gara, che abbiano pubblicato più di 5 dischi
- il numero totale di ospiti del festival
- il cantante più giovane che si è esibito in almeno due serate
- il numero medio per serata di cantanti in gara di genere pop