



## Prova Scritta del 15-09-2017

COGNOME \_\_\_\_\_ NOME \_\_\_\_\_ MATR \_\_\_\_\_

**Esercizio 1.** Interpretare la sequenza di 8 bit 10010110 come (a) numero naturale, (b) numero intero in modulo e segno e (c) numero intero in complemento a 2 e specificare per ognuno dei 3 casi a quale numero in base 10 corrisponde.

**Esercizio 2.** Scrivere un programma C++ che, dati in input un numero positivo  $N$  ed una matrice  $A$  di interi di dimensione  $m \times m$ , **utilizzando la ricorsione**, stampi OK se la somma dei valori presenti in ciascuno degli  $m$  "settori" della matrice  $A$  (vedi esempi) è strettamente minore di  $N$ , altrimenti stampi NO. Lo svolgimento corretto dell'esercizio senza l'utilizzo della ricorsione vale metà punteggio.

ESEMPIO 1: sia  $N = 27$  e sia  $A$  la seguente matrice  $5 \times 5$ , costituita quindi da 5 settori.

18	9	6	4	2
5	8	4	5	1
3	7	4	2	2
4	3	1	3	5
2	3	1	2	1

La somma dei valori presenti nel **setto**re 1 è 18.

La somma dei valori presenti nel **setto**re 2 è  $9 + 8 + 5 = 22$ .

La somma dei valori presenti nel **setto**re 3 è  $6 + 4 + 4 + 7 + 3 = 24$ .

La somma dei valori presenti nel **setto**re 4 è  $4 + 5 + 2 + 3 + 1 + 3 + 4 = 22$ .

La somma dei valori presenti nel **setto**re 5 è  $2 + 1 + 2 + 5 + 1 + 2 + 1 + 3 + 2 = 19$ .

Poiché **ciascuno** di questi numeri è strettamente minore di 27, verrà stampato OK.

**Esercizio 3.** (SOLO PER GLI STUDENTI IMMATRICOLATI PRIMA DEL 2015/2016) Si consideri il seguente programma, si descriva in maniera dettagliata il suo comportamento indicando cosa viene stampato in output. Non saranno considerate valide le risposte in cui è riportato il solo output senza adeguata motivazione.

```
#include<iostream>
int main()
{
  const int n = 6;
  char a[n];
  char *p = a;
  int i = 1;
  while ( p < a+n ){
    *p = 'z' - i;
    p++;
    i++; }
  char* q =a;
  p--;
  while(p > q)
    std::cout << *(q++)<<" "<<*(p--)<<std::endl;
  return 0;
}
```

**Esercizio 4.** (SOLO PER GLI STUDENTI DALL'A.A. 2015/2016 IN POI)

Si deve preparare un sistema per la gestione degli studenti delle università italiane.

A tale scopo si consideri la classe STUDENTE riportata di seguito che si può supporre interamente implementata.

```
class Studente {
  friend ostream& operator<<(ostream&, const Studente &);
private:
  string nome;
  string cognome;
  string codiceFiscale;

public:
  Studente ();
  Studente (const string&, const string&, const string&);
```



## Prova Scritta del 15-09-2017

COGNOME \_\_\_\_\_ NOME \_\_\_\_\_ MATR \_\_\_\_\_

```
const string& getNome() const;
const string& getCognome() const;
const string& getCodiceFiscale() const;
bool operator==(const Studente &) const;
};
```

Si consideri ora la classe UNIVERSITA, la cui interfaccia è riportata di seguito:

```
class Universita{
friend ostream& operator>>(ostream&, Universita &);
friend ostream& operator<<(ostream&, const Universita &);
private:
String nome;
Studente* studenti;
unsigned* matricole;
unsigned size;
unsigned capacity;

public:
Universita () _____ ;
Universita (Studente*, unsigned*) _____ ;
Universita (const Universita &) _____ ;
~ Universita () _____ ;
Universita & operator=(const Universita &) _____ ;
const String & getCodiceFiscaleStudente (unsigned matricola) _____ ;
void aggiuntiStudente(const Studente &, unsigned) _____ ;
bool isIscritto(const Studente &) _____ ;
const Studente & operator[](unsigned) _____ ;
unsigned getSize() _____ ;
```

\_\_\_\_\_ ; 2.a)

\_\_\_\_\_ ; 2.b)

```
};
```

1) Riportare a fianco ai metodi, dove opportuno, la parola chiave **const**.

2) Aggiungere intestazione ed implementazione di:

- un metodo per ottenere il nome dell'università
- un operatore -- (a scelta di pre o post-incremento) che riduca di 1 il valore di size

3) **Implementare opportunamente tutti i metodi** della classe Universita ad eccezione del metodo di lettura >> che si può considerare già implementato. (Nota: il metodo di stampa >> va implementato)

4) Si implementi inoltre un **main**, nel quale, letto da input un numero **N**, un array di **N** università ed una stringa **S**, stampi:

- Il numero di studenti aventi come nome **S**
- Lo studente iscritto a più università. A parità di iscrizioni stampare il primo che si incontra scorrendo l'array delle università.

Nota: è possibile modificare la classe Universita, ma solo per aggiungere eventuali metodi utili alla svolgimento del main. Se viene aggiunto qualche metodo è necessario riportarlo nell'intestazione della classe Universita sopra (sulla traccia) e presentare la corrispondente implementazione nel compito.

2/