

## Prova Scritta del 22-06-2017

**Esercizio 1.** La sequenza di bit 1101 1111 1110, nella rappresentazione in complemento a due a quale numero in base 10 corrisponde?

**Esercizio 2.** Scrivere una funzione booleana che, ricevuta come parametro una matrice  $X$  di interi di dimensione  $N \times M$ , utilizzando la ricorsione, restituisca true (risp. false) se esiste una riga  $i$  che soddisfa (risp. non soddisfa) la seguente proprietà: indicando con  $S$  la somma degli elementi contenuti nella riga  $i$ ,  $S$  è un elemento di ogni colonna della matrice  $X$ .

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

**Esercizio 3.** (SOLO PER GLI STUDENTI IMMATRICOLATI PRIMA DEL 2015/2016)

1) Si consideri la seguente dichiarazione

```
char* X[20];
```

Si indichino le istruzioni di allocazione necessarie affinché  $X$  sia una matrice dinamica di dimensione  $20 \times 10$ , e le corrispondenti istruzioni di deallocazione.

2) 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.

```
int main()
{
    unsigned s[10]={0}, *p=s+10;
    while(p!=s)
    {
        p-=2;
        *p=p-s;
    }
    for(unsigned i=0; i<10;i++)
        cout<<s[i]<<endl;
    return 0;
}
```

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

Si deve preparare un sistema per la gestione di una catena di negozi di strumenti musicali.

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

```
class Strumento {
friend ostream& operator>>(ostream&, Strumento &);
friend ostream& operator<<(ostream&, const Strumento &);
private:
    string modello;
```

**Prova Scritta del 22-06-2017**

```
    string tipo;
    float prezzo;
public:
    Strumento ();
    Strumento (const string&,const string&, float);
    const string& getModello() const;
    const string& getTipo() const;
    float getPrezzo() const;
    bool operator==(const Strumento &) const;
};
```

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

```
class Negozio{
friend istream& operator>>(istream&, Negozio &);
friend ostream& operator<<(ostream&, const Negozio &);
private:
    Strumento* strumenti;
    unsigned size;
    unsigned capacity;
public:
    Negozio ();
    Negozio (Strumento*, unsigned);
    Negozio (const Negozio &);
    ~ Negozio ();
    Negozio & operator=(const Negozio &);
    void aggiungiStrumento(const Negozio &);
    void rimuoviUltimoStrumentoInserito();
    float getSommaPrezzi() const;
    const Strumento & operator[](unsigned) const;
    void svuotaNegozio();
    const Strumento & getStrumentoPiuCostoso() const;

};
```

**Si implementino opportunamente tutti i metodi** della classe negozio ad **eccezione** del metodo di lettura >> che si può considerare già implementato.

Si implementi inoltre un main nel quale, letto da input un numero intero N ed un array di N negozi, stampi a) il negozio che vende lo strumento più costoso in assoluto, e b) il negozio più economico, ossia quello in cui la media dei prezzi di tutti gli strumenti è la minore in assoluto.

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