Prova Scritta del 21-07-2017

COGNOME_____**NOME**_____**MATR**_____

Esercizio 1. Si converta in complemento a due, utilizzando il minor numero di bit possibile, il numero $-c_3c_5$ dove c_3 e c_5 sono la terza e la quinta cifra del proprio numero di matricola.

Esercizio 2. Scrivere un programma C++ che, lette da input due frasi F1 e F2 di al più 100 caratteri, entrambe terminate dal carattere '.', verifichi se F1 e F2 sono frasi "simili", ossia se per ogni parola P1 di F1 esiste una parola P2 di F2 tale che **almeno** una delle seguenti condizioni è verificata:
1) P1 è uguale a P2,
2) P1 e P2 hanno la stessa lunghezza L e hanno in comune **più di** L/2 caratteri (non necessariamente nello stesso ordine),
3) P1 e P2 hanno la stessa lunghezza L, hanno in comune **esattamente** L/2 caratteri che occorrono in P1 e P2 nello stesso ordine.
Per la verifica di **UNA** delle 3 condizioni deve essere utilizzata la ricorsione. Le parole nelle frasi sono separate da spazi, virgole e punti interrogativi.
Ad Esempio, le frasi **F1**: *oggi vado in ufficio.* e **F2**: *dove andiamo oggi? a tuffare in piscina.* sono "simili", infatti per tutte le parole di F1 esiste una parola fi F2 per la quale è verificata una delle tre condizioni:
(P1=oggi, P2 =oggi, condizione 1) , (P1=vado, P2=dove, condizione 2), (P1=in, P2=in, condizione 1), (P1=ufficio, P2=tuffare, condizione 3).
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 istruzione di allocazione necessarie affinché X sia una matrice dinamica di dimensione 20*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()
  {
  int s[20], *p=s+20, i=-10;
  while(p!=s)
    {
    p--;
    *p=++i;
    }
  for(unsigned i=0; i<20;i++)
      cout<<s[i]<<endl;
  return 0;
  }
```


87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 - fax (+39) 0984 496410

Prova Scritta del 21-07-2017

COGNOME_____**NOME**_____**MATR**_____

Esercizio 4. (SOLO PER GLI STUDENTI DALL'A.A. 2015/2016 IN POI)
Si deve preparare un sistema per la gestione di una gelateria.
A tale scopo si consideri la classe GELATO riportata di seguito che si può supporre interamente implementata.

```
enum TipoGelato {CONO, COPPETTA, BRIOCHE};

class Gelato {
friend ostream& operator>>(ostream&, Gelato &);
friend ostream& operator<<(ostream&, const Gelato &);
private:
    string* gusti;
    unsigned numeroGusti;
    TipoGelato tipo;
public:
    Gelato ();
    Gelato (const string *,unsigned, TipoGelato);
    const string& getGusto(unsigned) const;
      unsigned getNumeroGusti() const;
      TipoGelato getTipo() const;
      void setTipo(TipoGelato);
      bool contieneGusto(const string &) const;
    bool operator==(const Gelato &) const;
};
```

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

```
class Gelateria{
friend istream& operator>>(istream&, Gelateria &);
friend ostream& operator<<(ostream&, const Gelateria &);
private:
    Gelato* gelatiInVendita;
    unsigned size;
    unsigned capacity;
public:
    Gelateria ();
    Gelateria (Gelato*, unsigned);
    Gelateria (const Gelateria &);
    ~Gelateria ();
    Gelateria & operator=(const Gelateria &);
    void aggiungiGelato(const Gelato &);
    unsigned getNumeroGelatiPerTipo(TipoGelato) const;
    const Gelato & operator[](unsigned) const;
    const Gelato & getGelatoConPiuGusti() const;
    void trasformaTuttiIConiInCoppette();
    const Gelato & getUltimoGelatoInserito() const;
};
```

Si implementino opportunamente tutti i metodi della classe gelateria ad **eccezione** del metodo di lettura >> che si può considerare già implementato. **Attenzione**: il metodo di stampa << va implementato.

Prova Scritta del 21-07-2017

COGNOME_____**NOME**_____**MATR**_____

Per il metodo `getGelatoConPiuGusti()` si restituisca il gelato con più gusti nella gelateria e se ce n'è più di uno restituire il primo che si incontra nell'array.

Si implementi inoltre un main nel quale, letto da input un numero intero N ed un array di N gelaterie, stampi:
a) la gelateria con più gelati, e a parità di gelati la prima che si incontra nell'array
b) il numero di gelaterie che hanno almeno un gelato di ogni tipo.
c) VERO se tutte le gelaterie hanno almeno un gelato con gusto "vaniglia", FALSO altrimenti.

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

CAMPUS DI ARCAVACATA http://www.unical.it/portale/strutture/dipartimenti_240/matinf/
87036 Arcavacata di Rende (CS) - Ponte Pietro Bucci Cubo 30B
tel. (+39) 0984 496402 - fax (+39) 0984 496410