

**NOTE SULLO SVOLGIMENTO DELLA PROVA SCRITTA:**

- **SPEGNERE I TELEFONINI;**
- **SCRIVERE IL PROPRIO NOME, COGNOME E NUMERO DI MATRICOLA SU OGNI FOGLIO UTILIZZATO;**
- **NON È POSSIBILE CONSULTARE NESSUN TIPO DI MATERIALE;**
- **NON È POSSIBILE UTILIZZARE CALCOLATRICI;**
- **PRIMA DI SCRIVERE LA SOLUZIONE DELL'ESERCIZIO, INSERIRE IL NUMERO DI ESERCIZIO CHE SI STA RISOLVENDO. PER ESEMPIO, SCRIVERE "ESERCIZIO N. 1" QUANDO SI STA RISOLVENDO L'ESERCIZIO N. 1;**
- **NON COPIARE DAL VICINO (NON È DETTO CHE IL VICINO SIA PIÙ BRAVO DI VOI);**
- **NON PERMETTETE AL VICINO DI COPIARE (È SPIACEVOLE VEDERSI ANNULLARE IL COMPITO SENZA COLPE);**
- **I PRIMI TRE ESERCIZI VALGONO 6 PUNTI; GLI ULTIMI 3 VALGONO 4 PUNTI.**
- **ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.**

1) Sia data la struttura seguente

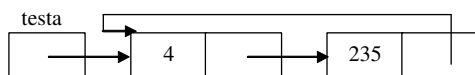
```
struct elem {char info; elem* pun;};
```

Scrivere una funzione ricorsiva che, data una lista di elementi di tipo elem, restituisca il numero di elementi con campo info uguale ad una lettera minuscola dell'alfabeto.

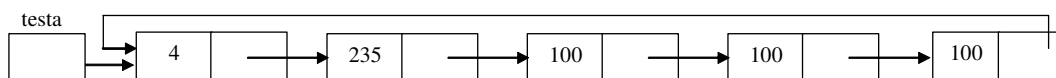
2) Sia data la struttura seguente

```
struct elem {int info; elem* pun;};
```

Scrivere una funzione che, data una lista circolare di elementi di tipo elem ed un intero positivo n, aggiunge n elementi in fondo alla lista. Tutti i nuovi elementi hanno campo informazione uguale a 100. In una lista circolare il campo pun dell'ultimo elemento punta al primo elemento della lista. Sia testa il puntatore al primo elemento della lista circolare. Se  $n = 3$ , la lista seguente:



viene modificata dalla funzione come segue:



3) Scrivere una funzione che, dati due vettori di interi  $v_1$  e  $v_2$  con elementi ordinati per valori crescenti, restituisce un nuovo vettore che contiene tutti gli elementi di  $v_1$  e di  $v_2$  ordinati per valori decrescenti. I vettori  $v_1$  e  $v_2$  ed il vettore risultato possono contenere duplicati.

Dati i due vettori  $v_1 = [2, 7, 21, 21, 45]$  e  $v_2 = [5, 45]$ ,

il vettore restituito dalla funzione è il seguente:  $[45, 45, 21, 21, 7, 5, 2]$ .

4) Data la rappresentazione  $(A09B)_{16}$  in base 16, trasformarla in base 2.

Scrivere la rappresentazione binaria in complemento a 2 su 8 bit del numero negativo -100.

5) Si mostri l'uscita a video del programma C++ seguente:

<pre>#include &lt;iostream&gt; using namespace std;  class alpha{ public:     int a;     alpha() { a=10; cout &lt;&lt; "alpha:a=" &lt;&lt; a &lt;&lt; endl;}     void stampa() { a++; cout &lt;&lt; "alpha:stampa() a=" &lt;&lt; a &lt;&lt; endl;}     virtual ~alpha() { cout &lt;&lt; " alpha:distr" &lt;&lt; endl; } };  class beta: public alpha{ public:     beta(int x = 1) { a=x; cout &lt;&lt; "beta:a=" &lt;&lt; a &lt;&lt; endl; }     virtual void stampa() { a++; cout &lt;&lt; "beta:stampa() a=" &lt;&lt; a &lt;&lt; endl; }     ~beta() { cout &lt;&lt; " beta:distr" &lt;&lt; endl; } };  class gamma: public beta{     beta obj;     int a; public:     gamma(int x) : beta(x) { a=x; cout &lt;&lt; "gamma:a=" &lt;&lt; a &lt;&lt; endl; }     void stampa() { a++ ; cout &lt;&lt; "gamma:stampa() a=" &lt;&lt; a &lt;&lt; endl; }     ~gamma() { cout &lt;&lt; "gamma :distr" &lt;&lt; endl; } };</pre>	<pre>int main(){     gamma* pc = new gamma(20);     alpha* pa=pc;     beta* pb =pc;     pa-&gt;stampa();     pb-&gt;stampa();     pc-&gt;stampa();     delete pa;     return 0; }</pre>
---	---

6) Si mostri l'uscita a video del programma C++ seguente, per i seguenti valori in ingresso:

1) x=10, y=10

2) x=20, y=5

3) x=20, y=10:

<pre>#include&lt;iostream&gt; using namespace std; class ecc { public:     int tipo;     ecc(int n) { tipo = n;                 cout &lt;&lt; "nuova eccezione" &lt;&lt; endl;} };  class ecc1: public ecc { public:     ecc1() : ecc(1) { cout &lt;&lt; "ecc1" &lt;&lt; endl; } }; class ecc2: public ecc { public:     ecc2() : ecc(2) { cout &lt;&lt; "ecc2" &lt;&lt; endl;} };  int g(int x) {     if (x&lt;=1) throw ecc1();     if (x&gt;10) throw ecc2();     if (x==10) throw 0;     return x; }</pre>	<pre>int f(int x, int y) {     try {         if (x&lt;=y) throw ecc1();         g(x-y);         return x+y;     }     catch (ecc2 &amp;e){ cout &lt;&lt; "ecc f: " &lt;&lt; e.tipo &lt;&lt; endl;}     return 0 ; }  int main(){     int x, y;     cin &gt;&gt;x &gt;&gt; y;     try{         cout &lt;&lt; f(x,y) &lt;&lt; endl;     }     catch (ecc &amp;e){ cout &lt;&lt; "ecc main: " &lt;&lt; e.tipo &lt;&lt; endl; }     catch (...){ cout &lt;&lt; "ecc generica"&lt;&lt; endl; }     return 0; }</pre>
--	---