

**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);
- TUTTI GLI ESERCIZI VALGONO 5 PUNTI
- ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.

1) Scrivere una funzione ricorsiva che dato un intero N produca la stampa seguente (nell'esempio, N = 5); nella definizione della funzione utilizzare il numero di argomenti ritenuti necessari.

```
* * * * *
* * * *
* * *
* *
*
```

2) Sia data la struttura seguente

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

Scrivere una funzione che, data una coda c di elementi di tipo elem, realizzata mediante una lista con puntatore al primo ed all'ultimo elemento, rimuove dalla coda l'elemento inserito per primo tra quelli il cui campo nome coincide con la stringa passata come argomento alla funzione. La funzione restituisce true se l'elemento viene trovato; false altrimenti. Per esempio, se nella coda rappresentata nella figura a sinistra viene chiamata la funzione passando la stringa "Franco", la funzione restituisce true e la coda diventa come in figura a destra.

```
Pippo
Franco
Mario
Franco
```

```
Pippo
Mario
Franco
```

3) Data una matrice di caratteri composta di r righe e c colonne, con r e c variabili, scrivere una funzione che per ogni riga ordini i caratteri per valori crescenti e stampi su uscita standard la matrice risultante. Nell'esempio seguente, la matrice a sinistra è la matrice originale e la matrice a destra è quella generata dalla funzione.

```
a b e d      a b d e
r s u a      a r s u
d y t s      d s t y
w c s a      a c s w
```

- 4) Data la rappresentazione  $(3\ 4\ 2)_5$  in base 5, trasformarla in base 9.  
 Data la rappresentazione in complemento a due  $(10010101)$ , esprimere il numero in base 10.

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

<pre>#include &lt;iostream&gt; using namespace std;  class A { public:     int a;     A() { a=0 ; cout &lt;&lt; a &lt;&lt; endl;}     A(int x) { a = x; cout &lt;&lt; a &lt;&lt; endl;} };  class C : public A { public:     C(): A(3) { a++ ; cout &lt;&lt; a &lt;&lt; endl;}     void f1() {cout &lt;&lt; 100 &lt;&lt; endl;} };  class D: public C { public:     D(){ a++ ; cout &lt;&lt; a &lt;&lt; endl;}     void virtual f1(){ cout &lt;&lt; 200 &lt;&lt; endl;} };</pre>	<pre>class E: public D { public:     E(){ cout &lt;&lt; 33 &lt;&lt; endl;}     void f1(){cout &lt;&lt; 300 &lt;&lt; endl;} };  template &lt;class T&gt; void fun(T * obj){     obj-&gt;f1(); }  int main(){     E* obj1= new E;     C* p1= obj1;     D* p2= obj1;     fun(p1);     fun(p2);     fun&lt;C&gt;(p2);     return 0; }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 6) Si mostri l'uscita a video delle esecuzioni del programma C++ seguente con, rispettivamente, gli ingressi  $x = 0$  e  $x = 1$ :

<pre>#include&lt;iostream&gt; using namespace std;  class A{ public:     int a;     A() {a = 5;} };  class AA: public A { public:     AA(int n) { a = a-n; } };  int fun1(int k) {     try {         if( k==0 )             throw AA(k);         k = 2;     }     catch(A&amp; obj) {         cout &lt;&lt; "Ecc. da fun1: " &lt;&lt; obj.a &lt;&lt; endl;     }     return k; }</pre>	<pre>int fun2(int j) {     try {         if (j)             throw A();         j += 3;     }     catch(AA&amp; obj) {         cout &lt;&lt; "Ecc. da fun2: " &lt;&lt; obj.a &lt;&lt; endl;     }     return j; }  int main(){     int x; cin &gt;&gt; x;     try {         cout &lt;&lt; fun1(x) &lt;&lt; endl;         cout &lt;&lt; fun2(x) &lt;&lt; endl;     }     catch (A&amp; obj) {         cout &lt;&lt; "Ecc. da main: " &lt;&lt; obj.a &lt;&lt; endl;     }     catch(...) {cout &lt;&lt; "Fine" &lt;&lt; endl;}     return 0; }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------