

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

```
#include <iostream>
using namespace std;
class A
{
protected:
    int i,j;
public:
    A(){ i = 0; j = 0; cout << i << ' ' << j << endl;}
    void virtual f() = 0;
    void stampa(){ cout << i << ' ' << j << endl;}
};

class B: public A
{
public:
    B(){ i++; j++; cout << i << ' ' << j << endl;}
    void f() { cout << i << ' ' << j << endl;}
};

class C: public B
{
protected:
    int z;
public:
    C(){ i++; j++; z = 2; cout << i << ' ' << j << ' ' << z << endl;}
    void f() { cout << i << ' ' << j << ' ' << z << endl;}
    void stampa(){ cout << i << ' ' << j << ' ' << z << endl;}
};

class D: public C
{
    int z;
public:
    D(){ i++; j++; z = 3; cout << i << ' ' << j << ' ' << z << endl;}
    void stampa(){ cout << i << ' ' << j << ' ' << z << endl;}
};
```

```
int main()
{
    cout << "(1)" << endl;
    D *d=new D;
    B *b=new B;
    A *a=d;
    cout << endl << "(2)" << endl;
    d->f();
    d->stampa();
    cout << endl << "(3)" << endl;
    a->f();
    a->stampa();
    cout << endl << "(4)" << endl;
    b->f();
    b->stampa();
    cout << endl << "(5)" << endl;
    C *c = new C;
    B *b1 = c;
    b1->f();
    b1->stampa();
    return 0;
}
```

2) Si mostri l'uscita a video delle esecuzioni del programma C++ seguente con, rispettivamente, gli ingressi:

- a) -1 1
- b) -1 -1
- c) -3 3
- d) 3 1
- e) 4 4

```
#include<iostream>
using namespace std;
class Ecc{
char e;
public:
Ecc(char c) { e=c; }
void print()
{
    if (e=='<') cout << "Neg" << endl;
    if (e=='>') cout << "Pos" << endl;
}
};

int f (int z)
{
    try
    {
        if (z>0) throw Ecc('>');
        if (z==-1) throw '-';
        if (z==0) throw false;
    }
    catch(Ecc e) {e.print();}
    return z+1;
}
```

```
int main()
{
    int x,y;
    cin >> x >> y;
    try
    {
        if (x>0 && y>0) throw Ecc('>');
        if (x<0 && y<0) throw Ecc('<');
        if (x==0 && y==0) throw false;
    }
    catch(Ecc e) {e.print();}
    catch(bool) { cout << "Zero" << endl;}
    x++; y++;
    try
    {
        cout << f(x) << endl;
        cout << f(y) << endl;
    }
    catch(bool) { cout << "Zero" << endl;}
    catch(...) {cout << "Fine" << endl;}
    return 0;
}
```

3) Scrivere la rappresentazione in base 2 dei numeri $(456)_{10}$ e $(6432)_{16}$.

Scrivere la rappresentazione in complemento a due (utilizzando 8 cifre) del numero intero -37 .

4) Scrivere una funzione che, date due stringhe S1 e S2, restituisce true se S1 è prefisso di S2 oppure S2 è prefisso di S1, false altrimenti. Una stringa X è prefisso di una stringa Y, se X corrisponde ai primi caratteri di Y. Per esempio, "pre" è prefisso di "precedente". SVOLGERE L'ESERCIZIO SENZA UTILIZZARE LE FUNZIONI DI LIBRERIA.

5) Sia data la struttura seguente

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

Scrivere una funzione che, data una lista l di elementi di tipo elem, ordinata per valori crescenti del campo info, restituisce una nuova lista con gli stessi elementi di l ordinati per valori decrescenti del campo info. La lista l NON DEVE ESSERE MODIFICATA.

6) Dati due vettori, v1 e v2, di interi ordinati per valori crescenti, scrivere una funzione che restituisce un vettore di interi ordinati per valori crescenti che contiene tutti gli elementi di v1 e v2. La funzione può essere chiamata con vettori di lunghezza diversa.