

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. I RIMANENTI TRE VALGONO 4 PUNTI.**
- **ALLA FINE DELLA PROVA, RICONSEGNARE TUTTI I FOGLI UTILIZZATI.**
- **SI RICORDA CHE LO SCRITTO VALE PER TRE APPELLI CONSECUTIVI (COMPRESO L'APPELLO IN CUI LO SCRITTO È SOSTENUTO)**

1) Scrivere una funzione ricorsiva che, dato un numero naturale N , restituisce la somma dei numeri pari inferiori o uguali ad N . Per esempio, se $N = 8$, la funzione restituisce 20 (che risulta dalla somma di $8+6+4+2$); se $N = 7$, la funzione restituisce 12 (che risulta dalla somma di $6+4+2$).

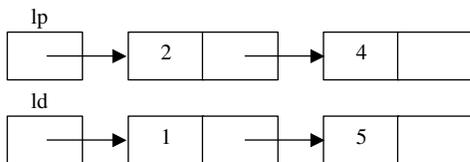
2) Sia data la struttura seguente

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

Scrivere una funzione che, data una lista l (non ordinata) di elementi di tipo `elem`, restituisce due liste lp e ld di elementi di tipo `elem` ordinate per valori crescenti del campo `info`; la lista lp contiene tutti gli elementi della lista l con campo `info` pari, mentre la lista ld contiene tutti gli elementi della lista l con campo `info` dispari. Per esempio, sia l la lista seguente:



Le liste lp e ld restituite dalla funzione sono rispettivamente:



3) Scrivere una funzione che, dati due vettori di interi $v1$ e $v2$ di dimensione N variabile, restituisce una matrice m di $N \times N$ interi, dove il generico elemento $m[i,j]=v1[i]+v2[j]$. Per esempio, dati i due vettori $v1 = [0,1,2]$ e $v2 = [3,4,5]$, la matrice m restituita dalla funzione è:

$$m = \begin{bmatrix} 3 & 4 & 5 \\ 4 & 5 & 6 \\ 5 & 6 & 7 \end{bmatrix}$$

4) Data la rappresentazione $(2AA17)_{16}$ in base 16, trasformarla in base 2.

Data la rappresentazione in complemento a due $(10001110)_{\text{comp}2}$, esprimere il numero in base 10.

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

```
#include <iostream>
using namespace std;

class A
{ protected:
  int aa;
public:
  A(int n=0) { aa=n; cout << "nuovo A: aa = " << aa << endl;}
};

class B
{ A a;
  protected:
  int x;
public:
  B(int n): a(n) { x=n; cout << "nuovo B: x = " << x << endl; }
  void f() { cout << "B::f() x=" << x << endl; }
};

class C: public B
{ A a;
public:
  C():B(1) { x++; cout << "nuovo C: x = " << x << endl; }
  virtual void f() { cout << "C::f() x=" << x <<endl; }
};

class D: public C
{ public:
  D() { x+=5; cout << "nuovo D: x = " << x << endl; }
  void f() { x+=10; cout << "D::f() x=" << x << endl; }
};
```

```
int main(){
  D d;
  C* pc = &d;
  pc->f();
  B* pb = pc;
  pb->f();
  return 0;
}
```

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

```
#include<iostream>
using namespace std;
template<class R, class S>
class A {
public:
  R f(R i, S d) {return static_cast<R>(i+d); }
};

template<class R, class S, class T>
T f (R r, S s, T t)
{ static int x = 1;
  A<R, S> a;
  cout << a.f (r, s) << endl;
  x++;
  return t+x;
}
```

```
int main ()
{ cout << f(2.6, 2.4, 1.1) << endl;
  cout << f<double,double,double>(1.0, 4, 1) << endl;
  cout << f<int,int>(4, 4.6, 1.1) << endl;
  return 0;
}
```