

ESERCIZI DI BASE (SEQUENZE)

<pre>/* File: ciao.c */ /* Scopo: il primo programma */ #include <stdio.h> int main(void) /* Stampa un messaggio sullo schermo. */ { printf("Ciao mondo!\n"); return 0; }</pre>	<pre>/* File: arearet1.c */ /* Scopo: uso di variabili */ #include <stdio.h> int main(void) /* Calcola l'area di un rettangolo di dimensioni fissate e la stampa. */ { int base; int altezza; int area; base = 3; altezza = 4; area = base * altezza; printf("Area: %d\n", area); return 0; } /* main */</pre>
<pre>/* File: arearet2.c */ /* Scopo: uso di istruzioni di ingresso e uscita */ #include <stdio.h> int main(void) /* Legge base e altezza di un rettangolo e ne stampa l'area. */ { int base, altezza, area; printf("Immetti base del rettangolo e premi INVIO\n"); scanf("%d", &base); printf("Immetti altezza del rettangolo e premi INVIO\n"); scanf("%d", &altezza); area = base * altezza; printf("Area: %d\n", area); return 0; } /* main */</pre>	<pre>/* File: divisione.c */ /* Scopo: uso di assegnazione e operatori aritmetici */ #include <stdio.h> int main(void) /* Calcola quoziente e resto di una divisione tra due interi. */ { int a, b; printf("Immetti due interi: "); scanf("%d%d", &a, &b); printf("%d : %d = %d con resto di %d\n", a, b, a/b, a%b); return 0; }</pre>
<pre>/* File: eurolire.c */ /* Scopo: uso di istruzioni di ingresso e uscita; uso di float */ #include <stdio.h> int main(void) /* Converte un valore da EURO in LIRE. */ { float euro;</pre>	

Semplici Esercizi. Prof. Fabio Cantaro
Materiale tratto da vari siti internet

```
printf("Immetti un Valore in EURO e premi INVIO\n");
scanf("%g", &euro);
printf("Valore in LIRE : %g\n", 1936.27 * euro);

return 0;
} /* main */
```

```
// limiti.cpp:
// stampa la lunghezza in byte di oggetti dei vari tipi
// e i relativi intervalli di definizione
//

#include <iostream.h>
#include <stdlib.h>
#include <limits.h> // per i limiti degli interi
#include <float.h> // per i limiti dei reali

int main()
{
    cout << "Un oggetto char occupa: sizeof(char) = " << sizeof(char)
        << " byte\n";
    cout << "Intervallo definizione char: CHAR_MIN = " << CHAR_MIN
        << "\t\tCHAR_MAX = " << CHAR_MAX << "\n\n";
    cout << "Un oggetto int occupa: sizeof(int) = "
        << sizeof(int) << "byte\n";
    cout << "Intervallo definizione int:\n\t INT_MIN = " << INT_MIN
        << "\t\tINT_MAX = " << INT_MAX << "\n\n";
    cout << "Un oggetto short occupa: sizeof(short) = "
        << sizeof(short) << " byte\n";
    cout << "Intervallo definizione short:\n\t SHRT_MIN = " << SHRT_MIN
        << "\t\tSHRT_MAX = " << SHRT_MAX << "\n\n";
    cout << "Un oggetto long occupa: sizeof(long) = "
        << sizeof(long) << " byte\n";
    cout << "Intervallo definizione long:\n\t LONG_MIN = " << LONG_MIN
        << "\t\tLONG_MAX = " << LONG_MAX << "\n\n";

    system("PAUSE");

    cout << "Un oggetto float occupa: sizeof(float) = "
        << sizeof(float) << " byte\n";
    cout << "Numero di cifre decimali significative float: FLT_DIG = "
        << FLT_DIG << '\n';
    cout << "Minimo e massimo esponente float:\n\t FLT_MIN_10_EXP = "
        << FLT_MIN_10_EXP << "\t\t FLT_MAX_10_EXP = " << FLT_MAX_10_EXP
        << "\n\n";
    cout << "Un oggetto double occupa: sizeof(double) = "
        << sizeof(double) << " byte\n";
    cout << "Numero di cifre decimali significative float: FLT_DIG = "
        << DBL_DIG << '\n';
    cout << "Minimo e massimo esponente double:\n\t DBL_MIN_10_EXP = "
        << DBL_MIN_10_EXP << "\t\t DBL_MAX_10_EXP = " << DBL_MAX_10_EXP
        << "\n\n";
    cout << "Un oggetto long double occupa: sizeof(long double) = "
        << sizeof(long double) << " byte\n";
    cout << "Numero di cifre decimali significative long double: LDBL_DIG = "
        << LDBL_DIG << '\n';
    cout << "Minimo e massimo esponente long double:\n\t LDBL_MIN_10_EXP = "
        << LDBL_MIN_10_EXP << "\t\t LDBL_MAX_10_EXP = " << LDBL_MAX_10_EXP
        << "\n\n";

    system("PAUSE");
```

```
cout << "Un oggetto unsigned int occupa: sizeof(unsigned int) = "  
    << sizeof(unsigned int) << " byte\n";  
cout << "Intervallo definizione unsigned int: 0, UINT_MAX = "  
    << UINT_MAX << "\n\n";  
cout << "Un oggetto unsigned short occupa: sizeof(unsigned short) = "  
    << sizeof(unsigned short) << " byte\n";  
cout << "Intervallo definizione unsigned int: 0, USHRT_MAX = "  
    << USHRT_MAX << "\n\n";  
cout << "Un oggetto unsigned long occupa: sizeof(unsigned long) = "  
    << sizeof(unsigned long) << " byte\n";  
cout << "Intervallo definizione unsigned long: 0, ULONG_MAX = "  
    << ULONG_MAX << "\n\n";  
  
system("PAUSE");  
return 0;}
```

```
/* File: tipocar.c */  
/* Scopo: istruzioni condizionali: Tipo di carattere in C */  
#include <stdio.h>
```

```
int main (void)  
{  
    char ch;  
  
    printf("Immetti un carattere: ");  
    scanf("%c", &ch);  
  
    if ('a' <= ch && ch <= 'z')  
        printf("lettera minuscola\n");  
    else if ('A' <= ch && ch <= 'Z')  
        printf("lettera maiuscola\n");  
    else  
        switch (ch) {  
            case '.': printf("punto\n"); break;  
            case ',': printf("virgola\n"); break;  
            case ';': printf("punto e virgola\n"); break;  
            case '!': printf("punto esclamativo\n"); break;  
            case '?': printf("punto interrogativo\n"); break;  
            case ':': printf("due punti\n"); break;  
            default: printf("altro carattere\n");  
        }  
  
    return 0;  
}
```

```
// limiti.cpp:  
// stampa la lunghezza in byte di oggetti dei vari tipi e i relativi intervalli di definizione  
//  
  
#include <iostream.h>  
#include <stdlib.h>  
#include <limits.h>    // per i limiti degli interi  
#include <float.h>     // per i limiti dei reali
```

```
int main()
{
    cout << "Un oggetto char occupa: sizeof(char) = " << sizeof(char)
        << " byte\n";
    cout << "Intervallo definizione char: CHAR_MIN = " << CHAR_MIN
        << "\t\tCHAR_MAX = " << CHAR_MAX << "\n\n";
    cout << "Un oggetto int occupa: sizeof(int) = "
        << sizeof(int) << "byte\n";
    cout << "Intervallo definizione int:\n\t INT_MIN = " << INT_MIN
        << "\t\tINT_MAX = " << INT_MAX << "\n\n";
    cout << "Un oggetto short occupa: sizeof(short) = "
        << sizeof(short) << " byte\n";
    cout << "Intervallo definizione short:\n\t SHRT_MIN = " << SHRT_MIN
        << "\t\tSHRT_MAX = " << SHRT_MAX << "\n\n";
    cout << "Un oggetto long occupa: sizeof(long) = "
        << sizeof(long) << " byte\n";
    cout << "Intervallo definizione long:\n\t LONG_MIN = " << LONG_MIN
        << "\t\tLONG_MAX = " << LONG_MAX << "\n\n";

    system("PAUSE");

    cout << "Un oggetto float occupa: sizeof(float) = "
        << sizeof(float) << " byte\n";
    cout << "Numero di cifre decimali significative float: FLT_DIG = "
        << FLT_DIG << "\n";
    cout << "Minimo e massimo esponente float:\n\t FLT_MIN_10_EXP = "
        << FLT_MIN_10_EXP << "\t\t FLT_MAX_10_EXP = " << FLT_MAX_10_EXP
        << "\n\n";
    cout << "Un oggetto double occupa: sizeof(double) = "
        << sizeof(double) << " byte\n";
    cout << "Numero di cifre decimali significative float: FLT_DIG = "
        << DBL_DIG << "\n";
    cout << "Minimo e massimo esponente double:\n\t DBL_MIN_10_EXP = "
        << DBL_MIN_10_EXP << "\t\t DBL_MAX_10_EXP = " << DBL_MAX_10_EXP
        << "\n\n";
    cout << "Un oggetto long double occupa: sizeof(long double) = "
        << sizeof(long double) << " byte\n";
    cout << "Numero di cifre decimali significative long double: LDBL_DIG = "
        << LDBL_DIG << "\n";
    cout << "Minimo e massimo esponente long double:\n\t LDBL_MIN_10_EXP = "
        << LDBL_MIN_10_EXP << "\t\t LDBL_MAX_10_EXP = " << LDBL_MAX_10_EXP
        << "\n\n";

    system("PAUSE");

    cout << "Un oggetto unsigned int occupa: sizeof(unsigned int) = "
        << sizeof(unsigned int) << " byte\n";
    cout << "Intervallo definizione unsigned int: 0, UINT_MAX = "
        << UINT_MAX << "\n\n";
    cout << "Un oggetto unsigned short occupa: sizeof(unsigned short) = "
        << sizeof(unsigned short) << " byte\n";
```

```
cout << "Intervallo definizione unsigned int: 0, USHRT_MAX = "  
    << USHRT_MAX << "\n\n";  
cout << "Un oggetto unsigned long occupa: sizeof(unsigned long) = "  
    << sizeof(unsigned long) << " byte\n";  
cout << "Intervallo definizione unsigned long: 0, ULONG_MAX = "  
    << ULONG_MAX << "\n\n";  
  
system("PAUSE");  
return 0;  
}
```

SELEZIONE (IF – ELSE)

<pre>/* File: maggiore.c */ /* Scopo: primo esempio sull'uso di if-then-else */ #include <stdio.h> int main(void) /* Legge due numeri e ne stampa il maggiore. */ { int primo, secondo, maggiore; printf("Primo numero ? "); scanf("%d", &primo); printf("Secondo numero ? "); scanf("%d", &secondo); if (primo > secondo) maggiore = primo; else maggiore = secondo; printf("maggiore: %d\n", maggiore); return 0; } /* main */</pre>	<pre>/* File: temperat.c */ /* Scopo: primo esempio di if-then-else in cascata */ #include <stdio.h> int main(void) /* Legge un valore reale che rappresenta una temperatura e stampa un opportuno messaggio. */ { float temperat; printf("Temperatura ? "); scanf("%g", &temperat); if (temperat > 30.0) printf("Molto caldo\n"); else if (temperat > 20.0) printf("Abbastanza caldo\n"); else if (temperat > 10.0) printf("Gradevole\n"); else if (temperat > 0) printf("Freddo\n"); else printf("Molto freddo\n"); return 0; } /* main */</pre>
<pre>/* File: mesesucc.c */ /* Scopo: uso di blocco di istruzioni */ #include <stdio.h> int main(void) /* Legge mese ed anno di una data e determina il mese successivo. */ { int mese, anno, mesesucc, annosucc; printf("Mese e anno ? "); scanf("%d%d", &mese, &anno); if (mese == 12) { mesesucc = 1; annosucc = anno + 1; } else { mesesucc = mese + 1; annosucc = anno; } printf("Il mese successivo al %d/%d e` %d/%d\n", mese, anno, mesesucc, annosucc);</pre>	<pre>/* File: triang3.c */ /* Scopo: uso di if-then-else con condizioni complesse */ #include <p2c/p2c.h> #include <stdio.h> int main(void) /* Legge le lunghezze dei lati di un triangolo e determina se il triangolo e' equilatero, isoscele o scaleno. Variante che utilizza if-then-else con condizioni complesse. */ { float primo, secondo, terzo; printf("Lunghezze lati triangolo ? "); scanf("%g%g%g", &primo, &secondo, &terzo); if (primo == secondo && primo == terzo) printf("Equilatero\n"); else if (primo == secondo primo == terzo secondo == terzo) printf("Isoscele\n"); else</pre>

<pre>return 0; } /* main */</pre>	<pre>printf("Scaleno\n"); return 0; } /* main */</pre>
<pre>// max3int.cpp // determina il massimo tra 3 interi// #include <iostream.h> #include <stdlib.h> int main () { int a,b,c; cout <<"a= "; cin >>a; cout <<"b= "; cin >>b; cout <<"c= "; cin >>c; if (a>b) { if (a>c) cout<<"a "<<a; else cout<<"c "<<c;} else {if (b>c) { if (b>c) cout<<"b "<<b;} else cout<<"c "<<c;} return 0 ; }</pre>	

SWITCH

<pre>/* File: giormese.c */ /* Scopo: uso dell'istruzione switch */ #include <stdio.h> int main(void) /* Legge un mese e determina il numero di giorni del mese. Non tiene conto di anni bisestili. */ { int mese, giorni_del_mese; printf("Mese (1..12) ? "); scanf("%d", &mese); switch (mese) { case 4: case 6: case 9: case 11: giorni_del_mese = 30; break; case 1: case 3: case 5: case 7: case 8: case 10: case 12: giorni_del_mese = 31; break; case 2:</pre>
--

```
    giorni_del_mese = 28;
    break;
}

printf("Giorni: %d\n", giorni_del_mese);
return 0;
} /* main */
```

```
/* File: datasucc.c */
/* Scopo: primo esempio significativo di uso di if-then-else annidati e
   istruzione switch */

#include <stdio.h>

int main(void)
    /* Legge giorno, mese ed anno di una data e determina la data del giorno
       successivo. */
{
    int giorno, mese, anno;
    int giorni_del_mese;

    /* lettura della data */
    printf("Data (giorno, mese, anno) ? ");
    scanf("%d%d%d", &giorno, &mese, &anno);
    printf("\nGiorno successivo al %d/%d/%d e` ", giorno, mese, anno);

    /* calcola il numero dei giorni del mese */
    switch (mese) {
    case 1: case 3: case 5: case 7: case 8: case 10: case 12:
        giorni_del_mese = 31;
        break;

    case 4: case 6: case 9: case 11:
        giorni_del_mese = 30;
        break;

    case 2:
        if (anno % 4 == 0 && anno != 1900)
            giorni_del_mese = 29;
        else
            giorni_del_mese = 28;
        break;
    }

    /* aggiorna la data del mese successivo */
    if (giorno == giorni_del_mese) {
        giorno = 1;
        if (mese == 12) {
            mese = 1;
            anno++;
        }
        else
            mese++;
    }
    else
        giorno++;

    /* stampa la data aggiornata */
    printf("%d/%d/%d\n", giorno, mese, anno);

    return 0;
}
```



```
} /* main */

// ascensor.cpp:
// simula un viaggio in ascensore tra piano interrato e attico(4° piano)
// segnala: AL PIANO - IN ARRIVO - SALITA - DISCESA - apre le porte
// CHIUDO PORTE - saliamo/scendiamo AL PIANO... - FINE DEL VIAGGIO
#include <iostream.h>
#include <stdlib.h>

int main()
{
    int partenza, posizione_iniziale, arrivo;
    enum {interrato, terra, primo, secondo, attico } piano;
    srand (333);
    cout << ".....VIAGGIO IN ASCENSORE.....\n\n";
    cout << " 0 --> INTERRATO ";
    cout << " 1 --> TERRA ";
    cout << " 2 --> PRIMO ";
    cout << " 3 --> SECONDO ";
    cout << " 4 --> ATTICO ";
    cout << "\n\n piano di partenza [0,1,2,3,4] = ";
    cin >> partenza; //un numero tra 0 e 4
    switch (partenza)
    {
        case 0: piano = interrato ; break;
        case 1: piano = terra ; break;
        case 2: piano = primo ; break;
        case 3: piano = secondo ; break;
        case 4: piano = attico ; break;
        default: cout << " valore non ammesso! per default stai al piano TERRA\n";
                piano = terra; partenza = 1; break;
    }
    posizione_iniziale = rand()*(5/(RAND_MAX+1.0));
    cout << "\n ascensore al piano ";
    switch (posizione_iniziale)
    {
        case 0: cout << " INTERRATO "; break;
        case 1: cout << " TERRA "; break;
        case 2: cout << " PRIMO "; break;
        case 3: cout << " SECONDO "; break;
        case 4: cout << " ATTICO "; break;
        default: cout << " TERRA \nvalore non ammesso! "
                << "\tper default l ascensore sta al piano TERRA";
                posizione_iniziale =1; break;
    }
    cout << " ("<<posizione_iniziale <<") ";
    if ( posizione_iniziale == partenza )cout << "PRESENTE\n";
    else {cout << " IN ARRIVO \n";
        if ( posizione_iniziale - partenza > 0) cout << " - DISCESA -\n";
        else cout << " - SALITA -\n ";}
```

```
int const larg =20;
int const t_max =10000000;
int i,j,t=0;
cout << "\tAPRO\t\t|-----|";
for (i=0;i<larg;i++) {cout << "\b\b\b|."; for (j=0;j<=t_max;j++)t++;};
cout << "\r\tAPERTA\n\n";
cout << " \tpuoi salire....\na quale piano vuoi andare? [0,1,2,3,4] = ";
cin >> arrivo;
if ( arrivo<0 || arrivo>4 )
    { arrivo =1;
      cout << "\nvalore non ammesso! "
            << "\tper default l ascensore va al piano TERRA\n" ;}
if ( arrivo == partenza )cout << "ci siamo gia'!!! ..... scendi! \n";
else {
    cout << " \nCHIUDO LE PORTE \n";
    if ( arrivo - partenza > 0) cout << " - saliamo ";
    else cout << " scendiamo ";
    cout << " al piano ";
    switch (arrivo)
        {
        case 0: cout << " INTERRATO\n "; break;
        case 1: cout << " TERRA\n "; break;
        case 2: cout << " PRIMO\n "; break;
        case 3: cout << " SECONDO\n "; break;
        case 4: cout << " ATTICO\n "; break;
        }
    }
cout << "\n\n .....apro le porte ..... FINE DEL VIAGGIO \n\n";

system("PAUSE");
```

CICLI

```
/*Trova tutti I numeri primi divisori interi di un numero intero N*/
#include <iostream.h>
#include <stdlib.h>

int main()
{
    int n, divisore;
    int trovato = 0; // indica se e` stato trovato un divisore

    cout << "Numero? ";
    cin >>n;

    divisore = 2;
    while ((divisore < n) && !trovato) {
        if ((n % divisore) == 0)
```

```
    trovato = 1;
else
    divisore++;
}

cout << "\nIl numero " << n;
if (trovato)
    cout << " non e` primo. E` divisibile per " << divisore << endl;
else
    cout << " e` primo.\n";

system("PAUSE");
return 0;
}
```

```
// risp.cpp:
// riconosce S o N

#include <iostream.h>
#include <stdlib.h>

int main()
{
    char risp;
    cout << "scrivi S o N: ";
    cin >> risp;
    while ((risp != 'N') && (risp != 'S'))
        { cout << "prova ancora; scrivi S o N: ";
          cin >> risp;
        }
    cout << "ultimo risp: " << risp << endl;
    system("PAUSE");
    return 0;
}
```

```
// quadrinf.cpp
#include <iostream.h>
#include <stdlib.h>

int main()
{
    int n, quadrato_n;
    n = 1;    //inizializzazione
    while (n <= 100)
        {
            quadrato_n = n*n;
            cout << n << ' ' << quadrato_n << '\t';
            //n=n+1; //incremento DIMENTICATO => LOOP INFINITO
        }
    cout << " n all'uscita del ciclo vale " << n << endl;
    system("PAUSE");
    return 0;
}
```

```
/* quadr.cpp
esegue il quadrato per i numeri da 1 a 100, VERSIONE CORRETTA*

#include <iostream.h>
#include <stdlib.h>

int main()
{
    int n, quadrato_n;
    n = 1;                //inizializzazione
    while (n<=100)
    {
        quadrato_n = n*n;
        cout<<n<<' '<<quadrato_n<<'\n';
        n=n+1;           //incremento
    }
    cout << " n all'uscita del ciclo vale " << n << endl;
    system("PAUSE");
    return 0;
}
```

```
/* MCD(a,B) con Alg.Euclide */
#include <iostream.h>
#include <stdlib.h>

int main()
{
    int a,b,n1,n2,i=0;
    cout << "\t\calcula MCD(a,B) con Alg.Euclide (differenza)\n"
        "\tcontando il N. di iterazioni [cfr.: mcdresto.cpp]\n\n"
        "introduci i due numeri a b ----->\n\ta= ";
    cin >>n1; cout<<"\tb= ";    cin >>n2;
    a=n1; b=n2;
    while (a!=b)
    { i++ ;
      if (a>b)
          {
              a=a-b;
          }
      else
          {
              b=b-a;
          }
    }
    cout << "iter.n° " <<i <<'\t';
    }
    cout << "\n\tMCD(" << n1 <<"," << n2 <<")=" << a <<'\n';

    system ("pause");
    return 0;
}
```

```
// lung1.cpp
```

```
/* Legge da tastiera una sequenza di interi [separati da spazio oppure invio] terminata da 0.
```

Calcola la lunghezza della sequenza (senza lo 0) e la stampa.

Versione che usa l'istruzione while. */

```
#include <iostream.h>
#include <stdlib.h>

int main()
{

int lunghezza; /* lunghezza della sequenza */
int dato; /* dato letto di volta in volta */

cout <<"Inserisci una sequenza di interi \n[separati da invio o spazio], terminata da 0\n";
lunghezza = 0; // all'inizio la lunghezza e' nulla
cin >>dato; // lettura del primo dato

while (dato != 0)
{
lunghezza++;
cin >>dato;
}
cout <<"La sequenza (senza lo 0 finale) e` lunga ..." <<lunghezza
<<" numeri" <<endl;

system("PAUSE");
return 0;
}
```

// lung2.cpp

/* Legge da tastiera una sequenza di interi [separati da spazio oppure invio] terminata da 0.

Calcola la lunghezza della sequenza (senza lo 0) e la stampa.

Versione che usa il l'istruzione do-while. */

```
#include <iostream.h>
#include <stdlib.h>

int main()
{

int lunghezza = 0; // all'inizio la lunghezza della sequenza e' nulla
int dato; /* dato letto di volta in volta */

cout <<"Inserisci una sequenza di interi \n" // lettura del primo dato
"[separati da invio o spazio], terminata da 0\n";
do {
cin >>dato;
lunghezza++;
} while (dato != 0);

cout <<"La sequenza (senza lo 0 finale) e` lunga ..." <<lunghezza-1
<<" numeri" <<endl;
```

```
system("PAUSE");  
return 0;  
}
```

```
// somma.cpp  
// Legge 10 interi, ne calcola la somma e la stampa.  
// cicli controllati da contatore
```

```
#include <iostream.h>  
#include <stdlib.h>  
  
int main()  
{  
    int conta, i, somma;  
  
    cout << "Immetti 10 interi: ";  
    somma = 0;  
    conta = 0;  
    while (conta < 10)  
    {  
        cin >> i;  
        somma += i;  
        conta++;  
    }  
    cout << "La somma e` " << somma << "\n";  
  
    system("PAUSE");  
    return 0;  
}
```

```
// pari.cpp  
// Legge N interi, e stampa i primi N numeri pari.  
// cicli controllati da contatore con limite letto in input
```

```
#include <iostream.h>  
#include <stdlib.h>  
  
int main()  
{  
    int conta, i=0;  
  
    cout << "Quanti numeri pari vuoi stampare: ";  
    cin >> conta;  
    while (conta > 0)  
    {  
        i = i + 2;  
        cout << i << ' ';  
        conta--;  
    }  
    cout << "\n";  
  
    system("PAUSE");  
    return 0;
```

```
}  
// sompone.cpp  
/* Legge una sequenza di numeri interi di lunghezza letta da tastiera.  
Stampa la somma dei numeri positivi e la somma dei numeri negativi.  
Utilizza un ciclo while. */  
  
#include <iostream.h>  
#include <stdlib.h>  
  
int main()  
{  
    int i;           // contatore di ciclo  
  
    int lung;       // lunghezza della sequenza  
  
    int numero;     // numero letto di volta in volta  
  
    int somma_pos, somma_neg; /* somme parziali degli interi positivi e  
                               negativi letti fino ad un certo punto */  
  
    cout <<"Immettere la lunghezza della sequenza di numeri interi!\n";  
    cin >>lung;  
    cout <<"Immettere "<<lung <<" numeri interi, uno per riga!\n" ;  
    somma_pos = 0;   /* prima di leggere il primo numero  
                     la somma parziale dei numeri positivi e` 0 */  
    somma_neg = 0;  /* prima di leggere il primo numero  
                     la somma parziale dei numeri negativi e` 0 */  
    i = 1;  
    while (i <= lung)  
    {  
        cin >>numero;  
        if (numero >= 0) somma_pos += numero;  
        else somma_neg += numero;  
        i++;  
    }  
    cout <<"Somma dei positivi: " << somma_pos <<endl;  
    cout <<"Somma dei negativi: " << somma_neg <<endl;  
  
    system("PAUSE");  
    return 0;  
}
```

```
// sompon2.cpp  
/* Legge una sequenza di numeri interi di lunghezza letta da tastiera.  
Stampa la somma dei numeri positivi e la somma dei numeri negativi.  
Utilizza un ciclo for. */  
  
#include <iostream.h>  
#include <stdlib.h>  
  
int main()  
{
```

```
int i;           // contatore di ciclo

int lung;       // lunghezza della sequenza

int numero;     // numero letto di volta in volta

int somma_pos, somma_neg; /* somme parziali degli interi positivi e
                           negativi letti fino ad un certo punto */

cout <<"Immettere la lunghezza della sequenza di numeri interi!\n";
cin >>lung;
cout <<"Immettere " <<lung <<" numeri interi, uno per riga!\n" ;
somma_pos = 0;   /* prima di leggere il primo numero
                 la somma parziale dei numeri positivi e` 0 */
somma_neg = 0;  /* prima di leggere il primo numero
                 la somma parziale dei numeri negativi e` 0 */
i = 1;

for (i = 1; i <= lung; i++)
{
    cin >>numero;
    if (numero > 0)  somma_pos += numero;
    else            somma_neg += numero;
}
cout <<"Somma dei positivi: " << somma_pos <<endl;
cout <<"Somma dei negativi: " << somma_neg <<endl;

system("PAUSE");
return 0;
}
```