

```
#include <iostream>

using namespace std;

void riempi(int a[], int dim) {
    for (int i=0; i<dim; i++) a[i]=i;
}

void stampa(int a[], int dim) {
    for (int i=0; i<dim; i++) cout << a[i];
}

int somma(int a[], int dim) {
    int tot=0;
    for (int i=0; i<dim; i++) tot +=a[i];
    return tot;
}
```

```
void riempi(int a[], int dim);  
void stampa(int a[], int dim);  
int somma(int a[], int dim);
```

```
#include <iostream>
#include <cmath>

using namespace std;

#include "array.h"

int main () {
    int val[10];
    double k;
    riempi(val,10);
    stampa (val,10);
    k = somma(val,10);
    k = pow((double)k, 10);
    cout << endl << k << endl;
    stampa (val,10);
}
```

```
#include <iostream>

using namespace std;

void fattori(int n) {
    int fattore = 2;

    while(n > 1) {
        int potenza = 0;

        while((n % fattore) == 0) {
            potenza++;
            n /= fattore;
        }
        if (potenza > 0) {
            cout << fattore;
            if (potenza > 1) cout << "^" << potenza;
            if (n > 1) cout << "*";
        }
        fattore++;
    }
    cout << endl;
}

int main() {
    int n;

    cout << "Inserire un numero: ";
    cin >> n;

    fattori(n);
}
```

```
#include <iostream>

using namespace std;

char lowerupper(char c) {
    if ((c >= 'a') && (c <= 'z')) {
        c = (c - 'a') + 'A';
    }
    else if ((c >= 'A') && (c <= 'Z')) {
        c = (c - 'A') + 'a';
    }
    return c;
}

int main() {
    while(!cin.eof()) {
        char c;

        cin.get(c);

        c = lowerupper(c);
        cout << c;
    }
}
```

```
#include <iostream>

// scommentare per utilizzare la versione di lettura che usa i limiti
// di sistema invece di usare la costante 1000.
// #define USELIMITS

#ifdef USELIMITS
// Inclusione condizionata alla definizione del simbolo USELIMITS
#include <limits> // per i limiti numerici
#endif

using namespace std;

int Max(int, int);
int Min(int, int);
double Average(int, int);

int read_integer(void);

int main() {
    int M = 0;
    int m = 0;
    int s = 0;

    for(int i = 0; i<3; i++) {
        int v;

        v = read_integer();
        // cout << "Inserire numero " << i << ": ";
        // cin >> v;
        M = Max(M, v);
        if (i == 0) {
            m = v;
        }
        else {
            m = Min(m, v);
        }
        s += v;
    }
    cout << "Valore massimo = " << M << endl;
    cout << "Valore minimo = " << m << endl;
    cout << "Valore medio = " << Average(s, 3) << endl;
}

int Max(int a, int b) {
    return (a > b) ? a : b;
}

int Min(int a, int b) {
    return (a > b) ? b : a;
}

double Average(int s, int n) {
    return (s / (double)n);
}

#ifdef USELIMITS
int read_integer(void) {
    bool correct = false;
    int x;

    while(correct == false) {
        cout << "Insert an integer: ";
        cin >> x;
        if (cin.fail() != 0) {
            cout << "You do not typed an integer. Try again" << endl;
        }
    }
}
```

```
    cin.clear();
    // Ignora 1000 caratteri oppure tutto quello prima di un '\n' se
    // '\n' viene incontrato prima di raggiungere il 1000esimo
    // carattere
    cin.ignore(1000, '\n');
}
else {
    correct = true;
}
}
return x;
}

#else
// versione alternativa usando i limiti del compilatore
int read_integer(void) {
    bool correct = false;
    int x;

    while(correct == false) {
        cout << "Insert an integer: ";
        cin >> x;
        if (cin.fail() != 0) {
            cout << "You do not typed an integer. Try again" << endl;
            cin.clear();
            // Ignora 1000 caratteri oppure tutto quello prima di un '\n' se
            // '\n' viene incontrato prima di raggiungere il 1000esimo
            // carattere
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
        else {
            correct = true;
        }
    }
    return x;
}
#endif
```

```
#include <iostream>

using namespace std;

int Max(int, int);
int Min(int, int);
double Average(int, int);

int main() {
    int M = 0;
    int m = 0;
    int s = 0;
    int n = 0;
    int v;

    cout << "Inserire numero " << n << ": ";
    cin >> v;

    while(v < 0) {
        M = Max(M, v);
        if (n == 0) {
            m = v;
        }
        else {
            m = Min(m, v);
        }
        s += v;
        n++;
        cout << "Inserire numero " << n << ": ";
        cin >> v;
    }
    if (n != 0) {
        cout << "Valore massimo = " << M << endl;
        cout << "Valore minimo = " << m << endl;
        cout << "Valore medio = " << Average(s, n) << endl;
    }
    else {
        cout << "Non sono stati inseriti numeri" << endl;
    }
}

int Max(int a, int b) {
    return (a > b) ? a : b;
}

int Min(int a, int b) {
    return (a > b) ? b : a;
}

double Average(int s, int n) {
    return (s / (double)n);
}
```



```
#include <iostream>

using namespace std;

void swap(int &, int &);

int main() {
    int v[3];

    for(int i = 0; i<3; i++) {
        cout << "Inserire numero " << i << ": ";
        cin >> v[i];
    }

    if (v[0] > v[1]) swap(v[0], v[1]);
    if (v[0] > v[2]) swap(v[0], v[2]);
    if (v[1] > v[2]) swap(v[1], v[2]);

    cout << "-----" << endl;
    for(int i = 0; i < 3; i++) {
        cout << "v[" << i << "] = " << v[i] << endl;
    }
    cout << "-----" << endl;
    for(int i = 2; i >= 0; i--) {
        cout << "v[" << i << "] = " << v[i] << endl;
    }
}

void swap(int & a, int & b) {
    int t = a;
    a = b;
    b = t;
}
```

```
#include <iostream>

using namespace std;

void swap(int v[], int i, int j);

int main() {
    int v[3];

    for(int i = 0; i<3; i++) {
        cout << "Inserire numero " << i << ": ";
        cin >> v[i];
    }

    if (v[0] > v[1]) swap(v, 0, 1);
    if (v[0] > v[2]) swap(v, 0, 2);
    if (v[1] > v[2]) swap(v, 1, 2);

    cout << "-----" << endl;
    for(int i = 0; i < 3; i++) {
        cout << "v[" << i << "] = " << v[i] << endl;
    }
    cout << "-----" << endl;
    for(int i = 2; i >= 0; i--) {
        cout << "v[" << i << "] = " << v[i] << endl;
    }
}

void swap(int v[], int i, int j) {
    int t = v[i];
    v[i] = v[j];
    v[j] = t;
}
```

```
#include <iostream>

using namespace std;

int main () {
    double d;
    int i;
    char c;
    int * pi = &i;

    delete pi;

    cout << "double = " << sizeof(d);
    cout << " add = " << &d << endl;
    cout << "int = " << sizeof(i);
    cout << " add = " << &i << endl;
    cout << "char = " << sizeof(c);
    cout << " add = " << &c << endl;

    return 0;
}
```

```
#include <iostream>

using namespace std;

int somma_ricorsiva(int n) {
    assert(n >= 0);
    if (n == 0) return 0;
    else
        return n + somma_ricorsiva(n-1);
}

int somma_iterativa(int n) {
    assert(n >= 0);
    int res = 0;
    for( ; n > 0; n--) {
        res = res + n;
    }
    return res;
}

int main() {
    int x;

    cout << "Inserire un intero positivo: ";
    cin >> x;
    cout << "La somma ricorsiva e': " << somma_ricorsiva(x) << endl;
    cout << "La somma iterativa e': " << somma_iterativa(x) << endl;
}
```

```
#include <iostream>

using namespace std;

void tree(int N) {
    int m = -1;
    for(int i = 1; i <= N; i++) {
        // per la riga i ci sono n-i spazi bianchi
        for(int j = 1; j <= N-i; j++) cout << ' ';
        m += 2;
        // per la riga i ci sono m asterischi (2 in piu' rispetto a i-1)
        for(int j = 1; j <= m; j++) cout << "**";
        cout << endl;
    }
}

int main() {
    int N;

    cout << "Inserire un numero >0 : ";
    cin >> N;

    if (N > 0) {
        tree(N);
    }
}
```