

```

/**
 * Fakultät einer Zahl
 */

#include <iostream>
using namespace std;

main()
{
    // Variablen bekannt machen
    int n, i, fakult;
    // Eingabe lesen
    cout << "Geben Sie eine pos. ganze Zahl ein: ";
    cin >> n;
    // Anfangswert
    fakult = n;
    // Fakultät berechnen
    // i läuft von n bis zur 1
    // (Vergleich mit Summenzeichen der Mathematik)
    if (n < 0)
        cout << "Neg. Zahlen nicht unterstuetzt!"
            << endl;
    else if (n == 0)
        fakult = 1;
    else {
        for (i=n-1; i>1; i=i-1)
        {
            fakult = fakult * i;
        }
        // Ausgabe des Ergebnis
        cout << n << "! = " << fakult << endl;
    }
    // fuer Windows, damit Fenster offen bleibt
    system("PAUSE");
}

```

```

/**
 * Kleinste natürliche Zahl Z für die gilt:
 *  $1 + 1/2 + 1/3 + \dots + 1/Z > G$  bei Eingabe von G
 * (G sollte nicht grösser als 10 sein)
 */

#include <iostream>
using namespace std;

main()
{
    // Variablen bekannt machen
    int g, z=0;
    float summe = 0.0;
    // Eingabe lesen
    cout << "Geben Sie eine pos. ganze Zahl ein: ";
    cin >> g;
    // Gültigkeit von G testen und dann approximieren
    if (g < 1)
        cout << "Zahl muss positiv sein!" << endl;
    else {
        while (summe <= g)
        {
            z = z + 1;
            summe = summe + (1.0/z);
            // Zwischenwerte ausgeben
            cout << summe << endl;
        }
        // Ausgabe des Ergebnis
        cout << "Grenze " << g << " bei z=" << z
            << " erreicht." << endl;
    }
    // fuer Windows, damit Fenster offen bleibt
    system("PAUSE");
}

```

```

/**
 * Feld der Länge 10 mit Werten füllen,
 * dann das grösste Element suchen und ausgeben.
 * "largest" speichert die Position im Feld.
 */

#include <iostream>
using namespace std;

main()
{
    // Variablen bekannt machen
    int feld[10], largest = 0, i;
    // Eingaben lesen
    for (i=1; i<=10; i++)
    {
        cout << i << ". Wert: ";
        cin >> feld[i];
    }
    // Grössten Wert suchen
    for (i=0; i<10; i++)
    {
        if (feld[i] > feld[largest])
            largest = i;
    }
    // grösstes Element ausgeben
    cout << "Groesstes Element an Position " << largest
         << " mit dem Wert " << feld[largest] << endl;
    // fuer Windows, damit Fenster offen bleibt
    system("PAUSE");
}

```