

```

cout << 4 << " " << (not(a < b) and not(N == 0)) << endl;
cout << 5 << " " << ((a or a)
                    and(a or not a)
                    and(not b or a)
                    and(not b or not a)
                    and(N == 0)) << endl;
cout << 6 << " " << (not(a > b) and N != 0) << endl;
cout << 7 << " " << (not((a > b) and N != 0))
    << endl;
cout << 8 << " " << (not((a > b) or N != 0))
    << endl;
cout << 9 << " " << (not(a > b) or not(N != 0)) << endl;
cout << 10 << " " << (not(a > b) and not(N != 0))
    << endl;
cout << 11 << " " << ((a > b) or N != 0) << endl;
cout << 12 << " " << (not(not a or b) and N == 0) << endl;
}

```

SOLUTION DE L'EXERCICE 21:

```

#include<iostream>

using namespace std;

int main()
{
    int a, b;

    cin >> a >> b;
    cout << (a and b) << endl;
}

```

11.2 Structures itératives

SOLUTION DE L'EXERCICE 22:

Imprime 123456

SOLUTION DE L'EXERCICE 23:

Imprime 123456

SOLUTION DE L'EXERCICE 24:

Imprime 123455

SOLUTION DE L'EXERCICE 25:

Imprime 1 2 3 4 5 5

SOLUTION DE L'EXERCICE 26:

```
#include <iostream>

using namespace std ;

int main(void){
    int n ;
    cin >> n ;
    for (int i=0; i<=n; ++i)
        cout<<i<<" " ;
    cout<<endl ;
}
```

SOLUTION DE L'EXERCICE 27:

```
#include<iostream>

using namespace std;

const int sentinelle = -1;

int main()
{
    int nombre = 0;
    double val, somme = 0;

    cin >> val;
    while (val != sentinelle) {
        somme += val;
        ++nombre;
        cin >> val;
    }
    cout << "Moyenne = " << (somme / nombre) << endl;
}
```

ou

```
#include<iostream>

using namespace std;

const int Sentinelle = -1;

int main()
{
    int nombre = 0;
    double val, somme = 0;
```

```
    for (cin >> val; val != Sentinelle; cin >> val) {
        somme += val;
        nombre++;
    }
    cout << "Moyenne = " << (somme / nombre) << endl;
}
```

SOLUTION DE L'EXERCICE 28:

```
int i;

i=a;
while(i<=b)
{
    <instruction;>
    i++;
}
```

et

```
int i;

i=a;
if(i<=b)
{
    do
    {
        <instruction;>
        i++;
    }
    while(i<=b);
}
```

SOLUTION DE L'EXERCICE 29:

```
#include<iostream>

using namespace std;

int main()
{
    int i, sum = 0;

    for (cin >> i; i > 0;) {
        i--;
        sum += 2 * i;
    }
    cout << sum << endl;
}
```

SOLUTION DE L'EXERCICE 30:

```
#include<iostream>

using namespace std;

int main()
{
    double prod;
    int i, cpt;

    prod = 1;
    cpt = 0;

    cin >> i;

    while (i < 100) {
        cpt++;
        prod *= i;
        i++;
    }
    cout << (prod / cpt) << endl;
}
```

SOLUTION DE L'EXERCICE 31:

```
#include<iostream>

using namespace std;

int main()
{
    int n, fact = 1;

    cin >> n;
    for (int i = 2; i <= n; i++) {
        fact *= i;
    }
    cout << fact << endl;
}
```

SOLUTION DE L'EXERCICE 32:

```
#include<iostream>

using namespace std;

const int Max = 100;

int main()
{
    int nombre = 0, val;
```

```
    cin >> val;
    while (val <= Max) {
        if (val < 0)
            nombre++;
        cin >> val;
    }
    cout << "Nombre = " << nombre << endl;
}
```

SOLUTION DE L'EXERCICE 33:

```
#include<iostream>

using namespace std;

int main()
{
    int n, i = 1;

    cin >> n;
    while (i * i <= n) {
        cout << i << endl;
        ++i;
    }
}
```

SOLUTION DE L'EXERCICE 34:

```
#include<iostream>

using namespace std;

int main()
{
    int n, i = 2;

    cin >> n;
    while (i < n) {
        cout << i << endl;
        i *= 2;
    }
}
```

SOLUTION DE L'EXERCICE 35:

```
#include<iostream>

using namespace std;

int main()
{
    int a, puiss;
```

```
    cin >> a;
    puiss = a;
    for (int i = 1; i <= 10; ++i) {
        cout << puiss << " ";
        puiss *= a;
    }
}
```

SOLUTION DE L'EXERCICE 36:

```
#include<iostream>

using namespace std;

int main()
{
    int b, somme, max;

    cin >> max >> b;
    somme = b;
    while (somme <= max and b != -1) {
        cout << b;
        cin >> b;
        somme += b;
    }
}
```

SOLUTION DE L'EXERCICE 37:

```
#include<iostream>

using namespace std;

int main()
{
    int bi, bs, i;

    cin >> bi >> bs;
    for (cin >> i; i < bi; cin >> i);
    for (; i <= bs; cin >> i) {
        cout << i;
    }
}
```

ou

```
#include<iostream>

using namespace std;

int main()
{
```

```
int bi, bs, i;

cin >> bi >> bs;
cin >> i;
while (i<bi)
    cin >> i;
while(i<=bs) {
    cout << i ;
    cin >> i ;
}

}
```

SOLUTION DE L'EXERCICE 38:

```
#include<iostream>

using namespace std;

int main()
{
    const int sentinelle = -1;
    int a, b, tmp;

    cin >> a;
    tmp = a;
    while (tmp != sentinelle) {
        b = tmp;
        cin >> tmp;
    }

    if (a == sentinelle) {
        cout << "La liste est vide";
    } else {
        if (a <= b) {
            cout << "A est inférieur à B";
        } else {
            cout << "A est strictement supérieur à B";
        }
    }
}
```

SOLUTION DE L'EXERCICE 39:

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

int main()
{
    int i, n, fiboA, fiboB, fiboC;
    cin >> n;
    i = 1;
```

```

    fiboA = 0;
    fiboB = 1;
    while (i < n) {
        fiboC = fiboA + fiboB;
        fiboA = fiboB;
        fiboB = fiboC;
        i++;
    }
    cout << fiboB << endl;
}

```

SOLUTION DE L'EXERCICE 40:

```

#include<iostream>

using namespace std;

// fiboA=F1, fiboB=F2, fiboC=F3;
// quotient_nouveau=fiboB/fiboA

int main()
{
    double eps, quotient_ancien, quotient_nouveau = 1.0, res;
    int fiboA = 1, fiboB = 1, fiboC = 2;

    cin >> eps;

    do {
        quotient_ancien = quotient_nouveau;
        quotient_nouveau = (double) fiboC / (double) fiboB;
        fiboA = fiboB;
        fiboB = fiboC;
        fiboC = fiboA + fiboB;

        res = quotient_ancien - quotient_nouveau;
        if (res < 0.0)
            res = -res;
    }
    while (res > eps);

    cout << "Nombre d'or = " << quotient_nouveau;
}

```

SOLUTION DE L'EXERCICE 41:

L'algorithme calcule r modulo d .

SOLUTION DE L'EXERCICE 42:

L'algorithme calcule a modulo n .

SOLUTION DE L'EXERCICE 43:

```
#include<iostream>

using namespace std;

int main()
{
    bool strictement = true, croissant = true, decroissant = true;
    int a, b;

    cin >> a >> b;

    /* Si on rencontre un terme plus petit que son successeur,
       la suite n'est pas décroissante ;
       si on rencontre un terme plus grand que son successeur,
       la suite n'est pas croissante ;
       si on rencontre un terme égal à son successeur,
       la suite n'est pas stricte.
    */

    while ((b != 0) and(croissant or decroissant)) {
        if (a < b) {
            decroissant = false;
        } else {
            if (a > b) {
                croissant = false;
            } else // a == b
            {
                strictement = false;
            }
        }
        a = b;
        cin >> b;
    }

    /* A la fin de l'analyse de la suite, on regarde quelles sont
       les variables booléennes qui sont vraies */

    if (croissant) {
        if (strictement) {
            cout << "La suite est strictement croissante";
        } else {
            cout << "La suite est croissante";
        }
    } else {
        if (decroissant) {
            if (strictement) {
                cout << "La suite est strictement decroissante";
            } else {
                cout << "La suite est décroissante";
            }
        } else {
            cout << "La suite est non triée";
        }
    }
}
```

```
}
```

SOLUTION DE L'EXERCICE 44:

```
#include<iostream>

using namespace std;

const double eps = 1.0e-7;
const int borne = 100;

int main()
{
    int val = 3, signe = 1, cpt = 0;
    double terme = 1.0;

    while ((1.0 / val) >= eps and cpt <= borne) {
        signe = -signe;
        terme += (double) signe / (double) val;
        val += 2;
        ++cpt;
    }
    cout << terme << endl;
}
```

SOLUTION DE L'EXERCICE 45:

```
#include<iostream>

using namespace std;

const double eps = 1.0e-7;
const int borne = 150;

int main()
{
    int val = 1, cpt = 0, denom;
    double terme = 0.0;

    do {
        denom = val * (val + 2);
        terme += 1.0 / denom;
        val += 4;
        ++cpt;
    }
    while ((1.0 / denom) >= eps and cpt <= borne);

    cout << terme << endl;
}
```

SOLUTION DE L'EXERCICE 46:

```
#include<iostream>

using namespace std;

const double eps = 1.0e-7;
const int borne = 120;

int main()
{
    int val = 2, denom;
    double terme = 1.0;

    do {
        denom = val * val;
        terme += 1 / (double) denom;
        ++val;
    }
    while ((1.0 / denom) >= eps and(val - 2) <= borne);

    cout << terme << endl;
}
```

SOLUTION DE L'EXERCICE 47:

```
#include<iostream>

using namespace std;

const double eps = 1.0e-8;

int main()
{
    int fact_n = 1, n = 1;
    double e = 0.0, terme = 1.0;

    while (2 * terme >= eps) {
        e += terme;
        fact_n *= n;
        terme = 1.0 / fact_n;
        ++n;
    }
    cout << e << endl;
}
```

SOLUTION DE L'EXERCICE 48:

```
#include<iostream>

using namespace std;

const double eps = 1.0e-5;
```

```
int main()
{
    double x, x2, terme, res, abs;
    int i;

    cin >> x;
    x2 = x * x;
    res = x;
    terme = -x * x2 / 6;

    if (terme < 0)
        abs = -terme;
    else
        abs = terme;

    i = 3;
    while (abs >= eps) {
        res += terme;
        terme *= -x2 / ((i + 1) * (i + 2));
        if (terme < 0)
            abs = -terme;
        else
            abs = terme;
        i += 2;
    }
    cout << res << endl;
}
```

SOLUTION DE L'EXERCICE 49:

```
#include <iostream>

using namespace std ;

int main(void)
{
    int n ;
    cin >> n ;
    for (int i=0; i<n; ++i) {
        for (int j=0; j<n; ++j) {
            cout<<"X" ;
        }
        cout<<endl ;
    }
    cout<<endl ;
}
```

SOLUTION DE L'EXERCICE 50:

```
#include <iostream>

using namespace std ;
```

```
int main(void) {
    int n ;
    cin >> n ;
    for (int i=0; i<=n; ++i) {
        for (int j=0; j<=n; ++j) {
            cout <<"("<<i<<","<<j<<")" ;
        }
        cout<< endl ;
    }
    cout<<endl ;
}
```

SOLUTION DE L'EXERCICE 51:

```
#include <iostream>

using namespace std ;

int main(void) {
    int n ;
    cin >> n ;
    for (int i=0; i<=n; ++i) {
        for (int j=0; j<=i; ++j) {
            cout <<"("<<i<<","<<j<<")" ;
        }
        cout<< endl ;
    }
    cout<<endl ;
}
```

SOLUTION DE L'EXERCICE 52:

```
#include <iostream>

using namespace std ;

int main(void)
{
    int n ;
    cin >> n ;
    // Le tracé des lignes supérieures
    // 1 2 3 4 5
    // XXXOXXXXXXOXXX
    for (int i=0; i<n/2; ++i) // i=num de ligne
    {
        // 1 tracer les X avant le 1er O
        for (int j=0; j<i; ++j)
            cout<< "X" ;
        // 2 le premier O
        cout<< "O" ;
        // 3 les X entres les 0
        for (int j=0; j<n-2*(i+1); ++j)
```

```

        cout<< "X" ;
        // 4 le second 0
        cout<< "0" ;
        // 5 les X de la fin
        for (int j=0; j<i; ++j)
            cout<< "X" ;
        cout<<endl ;
    }
    // Le tracé de la ligne "du milieu"
    for (int i=0; i<n/2; ++i)
        cout <<"X" ;
    cout<< "0" ;
    for (int i=0; i<n/2; ++i)
        cout <<"X" ;
    cout<< endl ;

    // Le tracé du bas: symétrique au haut.
    for (int i=n/2-1; i>=0; --i)
    {
        // 1 tracer les X avant le 1er 0
        for (int j=0; j<i; ++j)
            cout<< "X" ;
        // 2 le premier 0
        cout<< "0" ;
        // 3 les X entres les 0
        for (int j=0; j<n-2*(i+1); ++j)
            cout<< "X" ;
        // 4 le second 0
        cout<< "0" ;
        // 5 les X de la fin
        for (int j=0; j<i; ++j)
            cout<< "X" ;
        cout<<endl ;
    }
    cout<<endl ;
}

```

11.3 Les fonctions

SOLUTION DE L'EXERCICE 53:

```

#include<iostream>

using namespace std;

int exposant(int, int);

int main()
{
    int a, i;

    cin >> a >> i;
    cout << exposant(a, i) << endl;
}

```