1. **Thuật toán đệ quy**

* Giai thừa

**int** **giaithua**(int n)

{

    if (n == 0)

        return 1;

    return **giaithua**(n - 1) \* n;

}

* Fibonaci

**int** **fibo**(int n)

{

    if (n == 0)

        return 0;

    if (n == 1)

        return 1;

    return **fibo**(n - 1) + **fibo**(n - 2);

}

1. **Thuật toán đệ quy**

* Tháp Hà Nội

void **chuyen**(int n, char a, char b)

{

    cout **<<** a **<<** " -> " **<<** b;

    cout **<<** **endl**;

}

void **thap**(int n, char a, char b, char c)

{

    if (n == 1)

**chuyen**(1, a, c);

    else

    {

**thap**(n - 1, a, c, b);

**chuyen**(n, a, c);

**thap**(n - 1, b, a, c);

    }

}

* Xếp hậu

int n, buoc = 0;

int a[20];

bool **Ok**(int i, int j)

{

    for (int k = 1; k < i; k++)

        if (a[k] == j || **abs**(k - i) == **abs**(a[k] - j))

            return false;

    return true;

}

void **Dat**(int i)

{

    for (int j = 1; j <= n; j++)

    {

        if (**Ok**(i, j))

        {

            a[i] = j;

            if (i == n)

                buoc++;

            else

**Dat**(i + 1);

        }

    }

}

* Mã đi tuần

int X[8] = {2, 1, -1, -2, -2, -1, 1, 2};

int Y[8] = {1, 2, 2, 1, -1, -2, -2, -1};

int A[10][10] = {0}, n;

int dem = 0;

bool check = false;

void **in**()

{

    check = true;

    for (int i = 1; i <= n; i++)

    {

        for (int j = 1; j <= n; j++)

            cout **<<** A[i][j] **<<** " ";

        cout **<<** **endl**;

    }

}

void **move**(int x, int y)

{

    dem++;

    A[x][y] = dem;

    for (int i = 0; i < 8; i++)

    {

        if (dem == n \* n)

        {

**in**();

**exit**(0);

        }

        int u = x + X[i];

        int v = y + Y[i];

        if (u > 0 && u <= n && v > 0 && v <= n && A[u][v] == 0)

**move**(u, v);

    }

    dem--;

    A[x][y] = 0;

}

int **main**()

{

    cin **>>** n;

    int a, b;

    cin **>>** a **>>** b;

**move**(a, b);

    if (!check)

        cout **<<** "No solution!";

    return 0;

}

1. **Thuật toán quy hoạch động**

* Chuỗi con chung lớn nhất

int n;

cin **>>** n;

int a[1005], s[1005];

int res = 1;

for (int i = 0; i < n; i++)

    cin **>>** a[i];

for (int i = 0; i < n; i++)

{

    s[i] = 1;

    for (int j = 0; j < i; j++)

    {

        if (a[j] < a[i])

        {

            s[i] = **max**(s[j] + 1, s[i]);

        }

    }

     if (s[i] > res) res = s[i];

}

cout **<<** res;

* Xếp túi 0-1

int n, v;

int weight[1005], value[1005];

void **CaiTui**()

{

    int a[n + 1][v + 1];

    for (int i = 0; i <= n; i++)

    {

        for (int j = 0; j <= v; j++)

        {

            if (i == 0 || j == 0)

                a[i][j] = 0;

            else if (j < weight[i])

            {

                a[i][j] = a[i - 1][j];

            }

            else

            {

                a[i][j] = **max**(a[i - 1][j], value[i] + a[i - 1][j - weight[i]]);

            }

        }

    }

    cout << a[n][v] << **endl**;

}

1. **Danh sách liên kết đơn**

#include <bits/stdc++.h>

using namespace std;

*// 1. Khai báo*

struct **Student**

{

**string** name, id, loai;

    float point;

};

struct **node**

{

**Student** data;

**node** \*next;

};

**node** \***createNode**(**Student** val)

{

**node** \*temp;

    temp = new **node**();

    temp->next = **NULL**;

    temp->data **=** val;

    return temp;

}

*// Thêm vào đầu danh sách*

**node** \***addHead**(**node** \*head, **Student** val)

{

**node** \*temp = **createNode**(val);

    if (head == **NULL**)

        head = temp;

    else

    {

        temp->next = head;

        head = temp;

    }

    return head;

}

*// In danh sách*

void **printList**(**node** \*head)

{

**node** \*curr;

    curr = head;

    while (curr->next != **NULL**)

    {

        cout **<<** curr->data.id **<<** " - " **<<** curr->data.name **<<** " - " **<<** curr->data.point **<<** " - " **<<** curr->data.loai;

        cout **<<** **endl**;

        curr = curr->next;

    }

}

*// Tìm phần tử trong danh sách*

bool **findNode**(**node** \*head, **string** ids)

{

**node** \*curr;

    curr = head;

    while (curr->next != **NULL**)

    {

        if (curr->data.id **==** ids)

            return true;

        curr = curr->next;

    }

    return false;

}

*// Xóa phần tử trong sanh sách*

**node** \***deleteNode**(**node** \*head, **string** ids)

{

**node** \*curr, \*before;

    curr = head;

    before = **NULL**;

    while (curr->next != **NULL**)

    {

        if (curr->data.id **==** ids)

            break;

        else

        {

            before = curr;

            curr = curr->next;

        }

    }

    if (before == **NULL**)

    {

        head = curr->next;

    }

    else if (curr->next != **NULL**)

        before->next = curr->next;

    return head;

}

**string** **xepLoai**(float p)

{

    if (p <= 3.6)

        return "Yeu";

    if (p < 6.5)

        return "Trung binh";

    if (p < 7.0)

        return "Trung binh kha";

    if (p < 8.0)

        return "Kha";

    if (p < 9.0)

        return "Gioi";

    return "Xuat sac";

}

**node** \***inLoaiSV**(**node** \*head)

{

**node** \*curr;

    curr = head;

    while (curr->next != **NULL**)

    {

        curr->data.loai **=** **xepLoai**(curr->data.point);

        curr = curr->next;

    }

    return head;

}

void **lietke**(**node** \*head)

{

**node** \*curr;

    curr = head;

    while (curr->next != **NULL**)

    {

        if (curr->data.point >= 5)

        {

            cout **<<** curr->data.id **<<** " - " **<<** curr->data.name **<<** " - " **<<** curr->data.point **<<** " - " **<<** curr->data.loai;

            cout **<<** **endl**;

        }

        curr = curr->next;

    }

}

int **main**()

{

**string** s, ma;

    float x;

**Student** temp;

**node** \*head = new **node**();

*// 2. Nhập danh sách*

    do

    {

**getline**(cin, s);

**scanf**("\n");

**getline**(cin, ma);

        cin **>>** x;

        cin.**ignore**();

*// Thêm vào đầu danh sách*

        if (s **!=** "")

        {

            temp.name **=** s;

            temp.id **=** ma;

            temp.point = x;

            head = **addHead**(head, temp);

        }

    } while (s **!=** "");

*// 3. Tìm sinh viên trong danh sách*

    cout **<<** "Nhap ma sinh vien can tim: ";

    cin **>>** ma;

    if (**findNode**(head, ma))

        cout **<<** "Co trong danh sach\n";

    else

        cout **<<** "Khong tim thay\n";

*// 4. Xoa mot sinh vien*

    cout **<<** "Nhap ma sinh vien can xoa: ";

    cin **>>** ma;

    head = **deleteNode**(head, ma);

*// 5. Liệt kê sinh viên có điểm tb >=5*

    cout **<<** "Sinh viên có điểm trung bình >=5: ";

**lietke**(head);

*// 6. Xep loai sinh vien*

    head = **inLoaiSV**(head);

*// In danh sách*

    cout **<<** "Danh sách sinh viên: ";

**printList**(head);

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*Nguyen Van A*

*B20DCCN001*

*8.2*

*Le Thi B*

*B20DCCN002*

*9.6*

*Tran Duc C*

*B20DCCN003*

*7.8*

*Bui Thi D*

*B20DCCN004*

*3.6*

*B20DCCN000*

*0.0*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*