# ***Nhóm 9 \_ Bài tập chương 1 \_ Lý thuyết***

# ***Câu 1: Trong khoa học máy tính, CTDL được hiểu:***

* CTDL là cấu trúc (sự tổ chức) của DL/TT lên trên máy tính, mà ở đó với cấu trúc này máy tính có thể xử lý được.
* Cấu trúc này phải rõ ràng, xác định, các thành phần bên trong cấu trúc cũng phải rõ ràng, và phần xác định.
* Ví dụ:
  + CTDL cơ bản của một lớp học  
    (Mã lớp, tên lớp, tập sinh viên)
  + **Trong đó:**
    - Mã lớp, tên lớp có kiểu DL là **kiểu** **chuỗi.**
    - Tập sinh viên có **kiểu tập hợp** (tập hợp mà mỗi phần tử là một sinh viên).

# ***Câu 2: Trong khoa học máy tính, Giải thuật được hiểu:***

* GT là một tập hữu hạn của các bước (chỉ thị hay hành động) theo một trình tự, được xác định rõ ràng nhằm mục đích để giải quyết một bài toán nào đó (dựa vào những giá trị đầu vào gọi là **“input”** và cho ra kết quả đầu ra gọi là **“output”**)
* Ví dụ:
  + Trong kiến thức **Toán THCS**, ta có bài toán tìm no phương trình bậc hai một ẩn ***\*(1)***.
  + \*\*\*Ta có **giải thuật(T)** để giải bài toán tìm no cho phương trình ***\*(1):***
    - **Giải thuật (T);**
      * Đầu vào **(input)**: a, b, c (a, b, c )
      * Đầu ra **(output)**:kết luận nghiệm

# ***Câu 3:***

**Cấu trúc dữ liệu + Giải thuật = Chương trình**

Với một cấu trúc dữ liệu đã chọn, sẽ có những giải thuật tương ứng, phù hợp. Khi cấu trúc dữ liệu thay đổi thường giải thuật cũng phải thay đổi theo để tránh việc xử lý gượng ép, thiếu tự nhiên trên một cấu trúc không phù hợp. Hơn nữa, một cấu trúc dữ liệu tốt sẽ giúp giải thuật xử lý trên đó có thể phát huy tác dụng tốt hơn, vừa đáp ứng nhanh vừa tiết kiệm vật tư, giải thuật cũng dễ hiễu và đơn giản hơn.

**Ví dụ:** Một chương trình quản lý điểm thi của sinh viên cần lưu trữ các điểm số của 3 sinh viên. Do mỗi sinh viên có 4 điểm số ứng với 4 môn học khác nhau nên dữ liệu có dạng bảng như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i/j | j0 | j1 | j2 | j3 |
| i0 | 7 | 9 | 5 | 2 |
| i1 | 5 | 0 | 9 | 4 |
| i2 | 6 | 3 | 7 | 4 |

* **Phương án 1: Sử dụng mảng một chiều**

Có tất cả 3(SV)\*4(Môn) = 12 điểm số cần lưu trữ, do đó khai báo mảng result như sau:

int result [12] = {7, 9, 5, 2, 5, 0, 9, 4, 6, 3, 7, 4};

* + Với phương án này, thao tác xử lý được cài đặt như sau :  
    *void XuatDiem() //Xuất điểm số của tất cả sinh viên  
    {*   
    *const int so\_mon = 4;  
    int sv,mon;  
    for (int i=0; i<12; i+)  
    {  
    sv = i/so\_mon;  
    mon = i % so\_mon;  
    printf(“Điểm môn %d của sv %d là: %d”, mon, sv,result[i]);  
    }  
    }*
* **Phương án 2 : Sử dụng mảng 2 chiều**

Khai báo mảng 2 chiều result có kích thước 3 dòng\* 4 cột như sau:

int result [3][4] = {{7, 9, 5, 2},

{5, 0, 9, 4},

{6, 3, 7, 4}};

* + Với phương án này, thao tác xử lý được cài đặt như sau:  
    *void XuatDiem() //Xuất điểm số của tất cả sinh viên  
    {  
    int so\_mon = 4, so\_sv =3;  
    for ( int i=0; i<so\_sv; i+)  
    for ( int j=0; i<so\_mon; j+)  
    printf(“Điểm môn %d của sv %d là: %d”, j, i,result[i][j]);  
    }*

Nhận xét: Có thể thấy rõ phương án 2 cung cấp một cấu trúc lưu trữ phù hợp với dữ liệu thực tế hơn phương án 1, và do vậy giải thuật xử lý trên cấu trúc dữ liệu của phương án 2 cũng đơn giản, tự nhiên hơn.

# ***Câu 4:***

* For 🡪 O(n)  
  While 🡪 O(n)

🡺 while lồng trong for 🡺 O(n2)