**Câu Hỏi Chương 1**

**Câu 1:**

Cấu trúc dữ liệu là cấu trúc (sự tổ chức) của dữ liệu/thông tin lên trên máy tính, mà ở đó với cấu trúc này máy tính có thể sửa chữa được.

Vd: Cấu trúc dữ liệu cơ bản của 1 sinh viên (mã số sv, họ và tên, giới tính, ngày sinh, địa chỉ)

**Câu 2:**

Giải thuật là một tập hữu hạn 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 kết quả đầu ra gọi là “output”).

Vd: Bài toán tìm nghiệm ax + b = 0

Thuật giải:

Đầu vào (input): a, b (a, b ∈ R)

Đầu ra (output): kết luận nghiệm

B1: Nếu a = 0, b = 0 thì phương trình có vô số nghiệm

B2: Nếu a = 0, b ≠ 0 thì phương trình vô nghiệm

B3: Nếu a, b ≠ 0 thì nghiệm x = -b/a

**Câu 3:**

Vì giải thuật phản ánh các phép xử lý , còn đối tượng xử lý của giải thuật lại là dữ liệu, chính dữ liệu chứa đựng các thông tin cần thiết để thực hiện giải thuật. Như vậy giải thuật và cấu trúc dữ liệu có mối quan hệ chặt chẽ với nhau, được thể hiện qua công thức :

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

* Vd: Một chương trình quản lý điểm thi của sinh viên cần lưu trữ các điểm số của 3 sinh viên.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sinh viên** | **Môn 1** | **Môn 2** | **Môn 3** | **Môn 4** |
| **SV 1** | 7 | 9 | 5 | 2 |
| **SV 2** | 5 | 0 | 9 | 4 |
| **SV 3** | 6 | 3 | 7 | 4 |

* Khai báo a[][] có kích thước 3 dòng \* 4 cột:
* int a[3][4] ={{ 7, 9, 5, 2},{ 5, 0, 9, 4},{ 6, 3, 7, 4 }};
* Truy xuất điểm số môn j của sinh viên i - là phần tử tại (dòng i, cột j) trong bảng - cũng chính là phần tử nằm ở vị trí (dòng i, cột j) trong mảng.
* Bảngđiểm(dòng i, cột j) ⇒ a[i][j]
* void xuatDiem() //Xuất điểm số của tất cả sinh viên.

{

int r = 3, c = 4;

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

for (int j = 0; j < c; j++)

cout << a[i][j] << “/t”;

}

**Câu 4:**

Gọi a[j – 1] (\*) là phép toán cơ bản

Với i = 1, j chạy 1 lần => (\*) chạy 1 lần

Với i = 2, j chạy 2 lần => (\*) chạy 2 lần

Với i = 3, j chạy 3 lần => (\*) chạy 3 lần

….

Với i = n – 2, j chạy n – 2 lần => (\*) chạy n – 2 lần

Với i = n – 1, j chạy n – 1 lần => (\*) chạy n – 1 lần

* T(n) = (n – 1) + (n – 2) + (n – 3) + … + 2 + 1 = n2/2 + n/2 ≈ O(n2).