**Câu 1:**

* Trong khoa học máy tính, Cấu Trúc Dữ Liệu là cấu trúc (sự tổ chức) của dữ liệu/thông tin 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 xác định.

Ví dụ: Cấu trúc dữ liệu cơ bản của một sinh viên (mã số sv, họ và tên, giới tính, ngày sinh, địa chỉ).Trong đó: Mã số sinh viên, họ và tên, địa chỉ có kiểu dữ liệu là **kiểu chuỗi**; Ngày sinh của sinh viên có **kiểu Date** (kiểu ngày).

**Câu 2:**

Giải thuật 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: Giải phường trình bậc 2: ax2 + bx + c = 0

Input: a, b, c (a, b, c là số thực).

Output: Kết luận nghiệm

B1: Tính Delta

B2: Nếu Delta = 0, phương trình có nghiệm kép: x1 = x2 = -b/2a

Nếu Delta > = 0, phương trình có 2 nghiệm phân biệt:

x1 = (-b – (Delta))/2a; x2 = (-b + (Delta))/2a

Nếu Delta <0, phương trình vô nghiệm.

**Câu 3:**

* Các thành phần dữ liệu thực tế đa dạng, phong phú và thường chứa đựng những quan hệ nào đó với nhau, do đó trong mô hình tin học của bài toán, cần phải tổ chức , xây dựng các cấu trúc thích hợp nhất sao cho vừa có thể phản ánh chính xác các dữ liệu thực tế này, vừa có thể dễ dàng dùng máy tính để xử lý. Công việc này được gọi là xây dựng cấu trúc dữ liệu cho bài toán.
* Cấu trúc dữ liệu có thể được xem như là 1 phương pháp lưu trữ dữ liệu trong máy tính nhằm sử dụng một cách có hiệu quả các dữ liệu này. Và để sử dụng các dữ liệu một cách hiệu quả thì cần phải có các thuật toán áp dụng trên các dữ liệu đó. Do vậy, cấu trúc dữ liệu và giải thuật là 2 yếu tố không thể tách rời và có những liên quan chặt chẽ với nhau. Việc lựa chọn một cấu trúc dữ liệu có thể sẽ ảnh hưởng lớn tới việc lựa chọn áp dụng giải thuật nào.
* Từ những yêu cầu xử lý thực tế, cần tìm ra các giải thuật tương ứng để xác định trình tự các thao tác máy tính phải thi hành để cho ra kết quả mong muốn, đây là bước xây dựng giải thuật cho bài toán.
* Ví dụ:
* 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:**

-Tốt nhất

=> Số phép so sánh = n - 1

**-**Xấu nhất

Vòng lặp while: i -1 lần

Vòng lặp for:

|  |  |
| --- | --- |
| i = 1 | 0 |
| i = 2 | 1 |
| i = 3 | 2 |
| i = 4 | 3 |
| ….. | ….. |
| i = n – 2 | n – 3 |
| i = n – 1 | n – 2 |
| i = n | n - 1 |

* Số phép so sánh = (n(n – 1)) / 2