**Câu 1:**

* CTDL là cấu trúc tổ chức của dữ liệu/thông tin 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 xác định.

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

**Trong đó:**

* Mã số sv, 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: Ta có **giải thuật (T)** để giải quyết bài toán tìm nghiệm cho phương trình ax2  + bx + c = 0 như sau:

**Giải thuật (T)**:

Đầu vào (**input**): a, b, c (a, b, c )

Đầu rả (**output**): kết luận nghiệm

Bước 1: tính delta = b2  - 4ac

Bước 2: thực hiện kiểm tra delta

* 1. Nếu delta < 0 thì phương trình vô nghiệm;
  2. Nếu delta = 0 thì phương trình có nghiệm kép: x1  = x2  =
  3. Nếu delta > 0 thì phương trình có hai nghiệm phân biệt:

x1 =

x2  =

**Câu 3:**

* Giải thuật và cấu trúc dữ liệu có mối quan hệ mật thiết với nhau, được thể hiện qua công thức :

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

* CTDL và giải thuật là nền tảng cơ bản của ngành Công nghệ thông tin.

Ví dụ:

* Để biểu diễn các điểm số của sinh viên người ta dùng số thực thay vì chuỗi ký tự.
* Vì còn phải thực hiện thao tác tính trung bình từ những điểm số đó.

**Câu 4:**

Đếm số phép so sánh

* Số phép so sánh trong while
  + - Phép so sánh j > 0
      * Với i = 1 thì j chạy 2 lần => có 2 phép so sánh
      * Với i = 2 thì j chạy 3 lần => có 3 phép so sánh
      * Với i = 3 thì j chạy 4 lần => có 4 phép so sánh
      * …..
      * Với i =(n – 1)thì j chạy n lần => có n phép so sánh
    - Phép so sánh a[i+1] > x
      * Với i = 1 thì j chạy 1 lần => có 1 phép so sánh
      * Với i = 2 thì j chạy 2 lần => có 2 phép so sánh
      * Với i = 3 thì j chạy 3 lần => có 3 phép so sánh
      * …..
      * Với i = (n – 1) thì j chạy (n – 1) lần => có (n – 1)phép so sánh
  + Có n + (n – 1) = 2n phép so sánh
* Số phép so sánh trong for
  + - Phép so sánh i < n
  + Với n = 1 thì có 2n \* 1 + 1 phép so sánh
  + Với n = 2 thì có 2n \* 2 + 1 phép so sánh
  + Với n = 3 thì có 2n \* 3 + 1 phép so sánh
  + …..
  + Với n thì có 2n \* (n – 1) + 1 phép so sánh
* **T(n) = 2n \* (n – 1) + 1 O(n2 )**