***Câu 1***: Trong khoa học máy tính, ***Cấu trúc dữ liệu*** được hiểu như thế nào? Cho ví dụ.

* Cấu trúc dữ liệu là cấu trúc của dữ liệu/thông tin trên máy tính mà ở đó với cấu trúc này, máy tính có thể xử lý được.
* Ví dụ:

+ Cấu trúc dữ liệu cơ bản của 1 sinh viên:

* Trong đó: MSSV, họ và tên, địa chỉ có kiểu dữ liệu là kiểu chuỗi.
* Ngày sinh của sinh viên có kiểu Date.

***Câu 2***: Trong khoa học máy tính, giải thuật được hiểu như thế nào? Cho ví dụ.

* Giải thuật là một tập hữu hạn của các bước theo một trình tự, được xác định rõ ràng nhằm mục đích để giải quyết một bài toán nào đó.
* Ví dụ:

+ Trong kiến thức Toán THCS, ta có bài toán: Tìm nghiệm phương trình bậc 2 một ẩn có dạng ax2 +bx +c=0.

=>Ta có giải thuật để giải bài toán tìm nghiệm cho phương trình trên như sau:

Giải thuật:

* Đầu vào: a, b, c.
* Đầu ra: Kết luận nghiệm.

***Câu 3***: Tại sao nói CTDL và GT có quan hệ mật thiết vơi nhau? Liệt kê 1 ví dụ nói về cách thiết kế cấu trúc dữ liệu sẽ ảnh hưởng đến giải thuật, giải thích tại sao?

* Thực hiện một đề án tin học là chuyển bài toán thực tế thành bài toán có thể giải quyết trên máy tính. Một bài toán thực tế bất kỳ đều bao gồm các đối tượng dữ liệu và các yêu cầu xử lý trên những đối tượng đó.

Cấu trúc dữ liệu + Chương trình = Giải thuật

* Bản thân các phần tử của dữ liệu thường có môi quan hệ với nhau, ngoài ra nếu biết tổ chức chúng theo cấu trúc dữ liệu thích hợp thì việc thực hiện các phép xử lý trên các dữ liệu sẽ càng thuận lợi hơn, đạt hiệu quả cao hơn. Với 1 ctdl đã chọn ta sẽ có giải thuật xử lý tương ứng. Ctdl thay đổi thì giải thuật cũng thay đổi theo.

VD:

***Câu 4***: Đếm số phép so sánh trong giải thuật ở ví dụ 1.12.

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).