**TRẢ LỜI CÂU HỎI – CHƯƠNG 1**

**CÂU 1:**

Trong khoa học máy tính ,cấu trúc dữ liệu được hiểu là một cách lưu dữ liệu trong máy tính sao cho nó có thể được sử dụng một cách hiệu quả.

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

*Trong đó: ( mã số SV, họ và tên , giới tính, địa chỉ ) là* ***kiểu chuỗi****. (ngày sinh) là* ***kiểu date.***

**CÂU 2:**

Giải thuật là 1 tập hữu hạn của các bước theo một trình tự , được xác định rõ ràng nhằm mục đích giải quyết một bài toán nào đó(dự vào những giá trị input và output).

**Ví dụ**: Tìm nghiệm phương trình bậc nhất :  P(x): *a*x + *b* = *c*, (*a*, *b*, *c* là các [số thực](https://vi.wikipedia.org/wiki/S%E1%BB%91_th%E1%BB%B1c))

Giải thuật (T)

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

**Đầu ra(output):** kết luận nghiệm

**CÂU 3:**

Thực hiện một đề án tin học là chuyển bài toán thực tế thành bài toán có thể giải quyết trên máy tính. Mổ bài toán thực tế bất kì đều bao gồm các đối tượng dữ liệu và các yêu cầu xử lí trên những đối tượng đó. Vì thế để xây dựng một mô hình tin học phản ánh được bài toán thực tế cần chú trọng đến hai vấn đề:

1. Tổ chức biểu diễn các đối tượng thực tế:

•Các thành phần dữ liệu thực tế đa dạng , phong phú và thường chứa đựng những quan hệ nao đó với nhau, do đó trong mô hình tin học của bài toán cần phải có 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 lí. Công việc này được gọi là xây dựng cấu trúc dữ liệu cho bài toán.

1. Xây dựng các thao tác xử lí dữ liệu

•Khi giải quyết một bài toán trên máy tính, chúng ta chỉ có khuynh hướng chỉ chú trọng đến việc xây dựng giải thuật mà quên đi tầm quan trọng của việc tổ chức dữ liệu trong bài toán.

•Giải thuật phản ánh các phép xử lí, còn đối tượng của giải thuật lại là dữ liệu,chính dữ liệu chứa đựng các thông tin cần thiết để thực hiện giải thuật. Để xác định được giải thuật phù hợp cần thiết để nó tác động đến loại dữ liệu nào ( ví dụ để làm nhuyễn các hạt đậu, người ta dung cách xay chứ không băm bằng dao, vì đậu sẽ văng ra ngoài) và khi chọn lựa cấu trúc dữ liệu cũng cần phải hiểu rõ những thao tác nào sẽ tác động đến nó(ví dụ để biểu diễn các điểm số của sinh viên người ta dùng số thực thay vì chuỗi kí tự vì còn phải thực hiện thao tác tính trung bình từ những điểm số đó).

🡺Như vậy một đề án tin học giải thuật và cấu trúc dữ liệu có mối quan hệ mất thiết với nhau được thể hiện qua công thức sau:

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 thương giải thuật cũng thay đổi theo để tránh việc xử lí gượng ép, thiếu tự nhiên trên một cấu trúc không phù hợp.

**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:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sinh viên | Môn1 | Môn2 | Môn3 | Môn4 |
| Sinhvien1 | 7 | 9 | 5 | 2 |
| Sinhvien2 | 5 | 0 | 9 | 4 |
| Sinhvien3 | 6 | 3 | 7 | 4 |

Chỉ xét thao tác xử lý là xuất điểm số các môn của từng sinh viên.

Giả sử có các phương án tổ chức lưu trữ sau:

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};

khi đó trong mảng *result* các phần tử sẽ được lưu trữ như sau:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |

Sinh vien 1 sinh vien 2 sinh vien 3

Và truy xuất điểm số môn j của sinh viên i - là phần tử tại (dòng i, cột j) trong bảng - phải sử dụng một công thức xác định chỉ số tương ứng trong mảng result:

bảngđiểm(dòng i, cột j) ⇒ result[((i-1)\*số cột) + j]

Ngược lại, với một phần tử bất kỳ trong mảng, muốn biết đó là điểm số của sinh viên nào, môn gì, phải dùng công thức xác định sau

result[ i ] ⇒ bảngđiểm (dòng((i / số cột) +1), cột (i % số cột) )

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 }};

khi đó trong mảng *result* các phần tử sẽ được lưu trữ như sau :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cột 0 | Cột 1 | Cột 2 | Cột 3 |
| Dòng 0 | Result[0]=7 | Result[0]=9 | Result[0]=5 | Result[0]=2 |
| Dòng 1 | Result[0]=5 | Result[0]=0 | Result[0]=9 | Result[0]=4 |
| Dòng 2 | Result[0]=6 | Result[0]=3 | Result[0]=7 | Result[0]=4 |

Và 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) ⇒ result[ i] [j]

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]);

}

**Câu 4:**

Đếm số phép so sánh của vòng lập for bên trong:

|  |  |
| --- | --- |
| Lần | Số phép so sánh |
| Lần i=1 | 1+2 |
| Lần i=2 | 1+4 |
| ………… | ………… |
| Lần i=n-1 | 1+2(n-1) |

Đếm số phép toán so sánh:

|  |  |
| --- | --- |
| Lần | Số phép so sánh |
| Lần n=1 | 1 |
| Lần n=2 | 1+1\*(1+2(n-1)) |
| Lần n=3 | 1+2\*(1+2(n-1)) |
| ………… | ………… |
| Lần n=n | 1+(n-1)\*(1+2\*(n-1)) |

Vậy phép so sánh có 1+(n-1)\*(1+2\*(n-1))