**Câu 1:**

**Danh sách đặc** là danh sách mà các phần tử trong danh sách có *cùng kiểu dữ liệu* và được *cấp phát liên tục* trong bộ nhớ.

Ví dụ: Danh sách có 10 phần tử kiểu ***int***:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 6 | 9 | 7 | 10 | 15 | 4 | 2 | 1 | 5 |

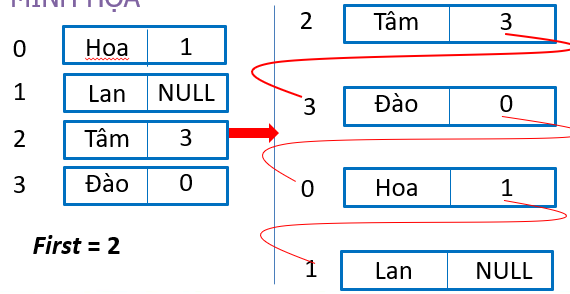
**Câu 2:**

**Danh sách liên kết** (Linked List) là một dãy các cấu trúc dữ liệu được kết nối với nhau thông qua các liên kết (link). Hiểu một cách đơn giản thì danh sách liên kết là một cấu trúc dữ liệu bao gồm một nhóm các nút (node) tạo thành một chuỗi. Mỗi nút gồm dữ liệu ở nút đó và tham chiếu đến nút kế tiếp trong chuỗi.

Theo chương trình học thì danh sách liên kết có loại là: danh sách liên kết đơn và danh sách liên kết kép. Tuy nhiên có rất nhiều loại danh sách liên kết khác.

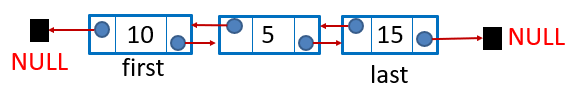
*Danh sách liên kết đơn*: chỉ duyệt các phần tử theo một chiều. Bao gồm 2 thành phần là link và info:

**Ví dụ**:



*Danh sách liên kết đôi***:** các phần tử có thể được duyệt theo chiều về trước hoặc về sau. Bao gồm 3 thành phần: previous, info, next

**Ví dụ:**

****

**Câu 3:**

STACK và QUEUE là danh sách hạn chế vì việc thêm vào và lấy ra 1 phần tử nào đó phải có quy tắc chứ không tùy tiện.

Ví dụ:

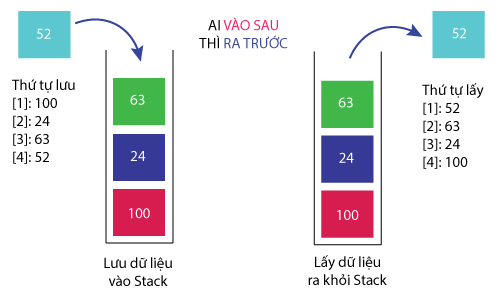
STACK: Phần tử nào được thêm vào sau sẽ được lấy ra trước.

QUEUE: Phần tử nào được thêm vào sau sẽ được lấy ra sau.

**Câu 4:**

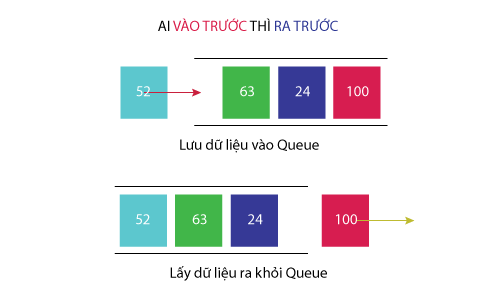
**LIFO**( Last In First Out): Phần tử được *thêm vào* ngăn xếp *sau*, sẽ *được lấy* ra (xóa) khỏi ngăn xếp *trước.*

Ví dụ:



**FIFO**(First In First Out): Phần tử được *thêm vào* trước, sẽ *được lấy* ra (xóa) *trước.*

Ví dụ:



**Câu 6:**

STACK: Trong trình biên dịch (thông dịch), khi thực hiện các thủ tục, Stack được sử dụng để lưu môi trường của các thủ tục. Trong một số bài toán của lý thuyết đồ thị (như tìm đường đi), Stack cũng thường được sử dụng để lưu dữ liệu khi giải các bài toán này.

QUEUE: Bộ đệm (ví dụ: Nhấn phím -> Bộ đệm -> CPU xử lý).Xử lý các lệnh trong máy tính (ứng dụng trong HÐH, trình biên dịch), hàng đượi các tiến trình chờ được xử lý,...

**Câu 7:**

**Cấu trúc dữ liệu động**: không được khai báo tường minh, không có tên gọi, xin khi cần, giải phóng khi sử dụng xong, được cấp phát trong heap, linh động về kích thước.

Ví dụ:

Int \*a;

a = new int => cấp phát bộ nhớ cho biến a kiểu int;

\*a = 10 => gán giá trị bằng 10 cho biến động a;