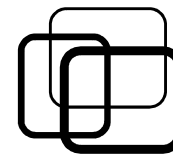


Ngăn xếp và Hàng đợi

GV. Nguyễn Minh Huy

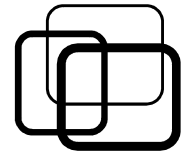


- Tổng quan về ngăn xếp.
- Tổng quan về hàng đợi.



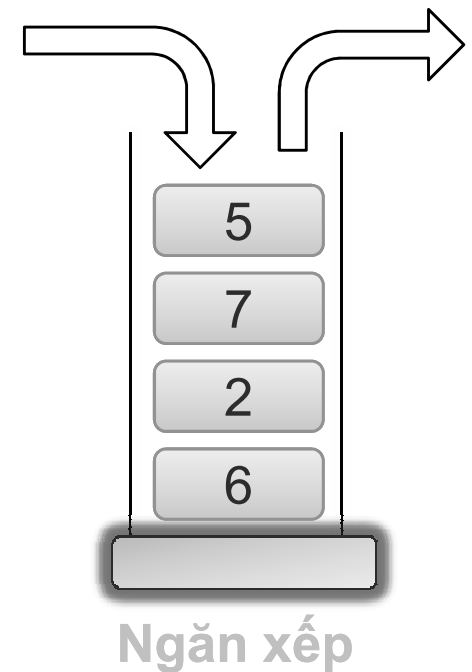
- **Tổng quan về ngăn xếp.**
- **Tổng quan về hàng đợi.**

Tổng quan về ngăn xếp

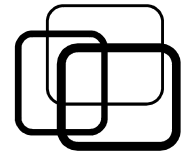


■ Khái niệm ngăn xếp:

- Dãy phần tử hoạt động theo cơ chế LIFO.
- Cơ chế LIFO (**L**ast **I**n **F**irst **O**ut).
 - Thứ tự vào sau ra trước.

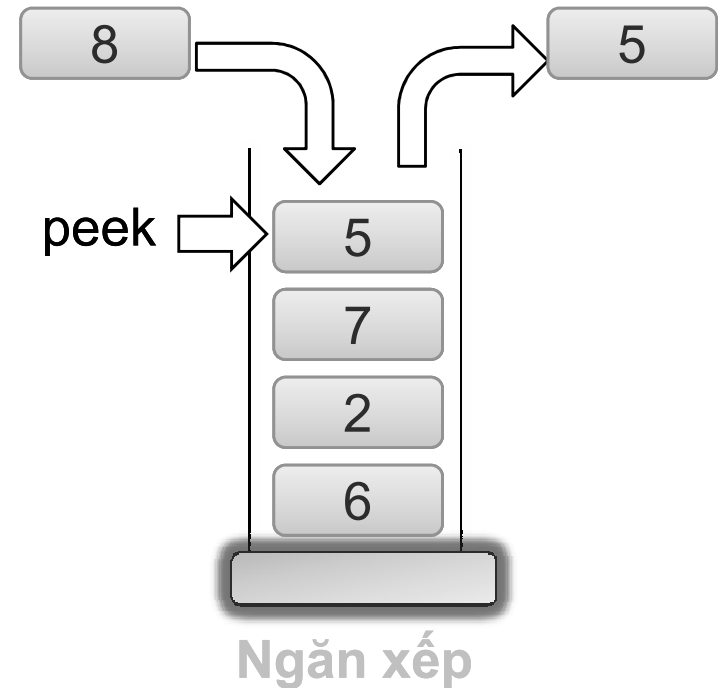


Tổng quan về ngăn xếp



■ Các thao tác trên ngăn xếp:

- push: thêm phần tử.
- pop: lấy phần tử.
- peek: đọc phần tử.
- init: khởi tạo.
- empty: kiểm tra rỗng.
- full: kiểm tra đầy.



Tổng quan về ngăn xếp



■ Cài đặt ngăn xếp trong C:

■ Khai báo:

// Dùng mảng động.

```
struct Stack
{
    int *data;
    int top;
    int size;
};
```

// Dùng danh sách liên kết đơn.

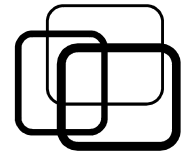
```
struct Stack
{
    SNode *top;
};
```

Tổng quan về ngăn xếp



- Cài đặt ngăn xếp trong C:
 - init.
 - isEmpty.
 - isFull.

Tổng quan về ngăn xếp

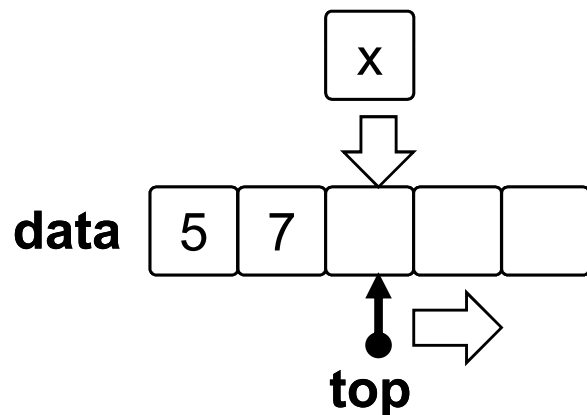


■ Cài đặt ngăn xếp trong C:

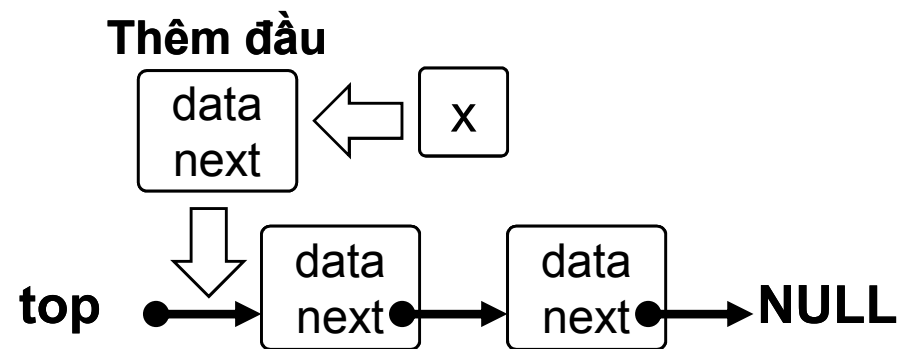
■ Các thao tác:

- Thêm phần tử: push.

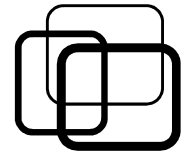
Dùng mảng động



Dùng danh sách liên kết đơn



Tổng quan về ngăn xếp

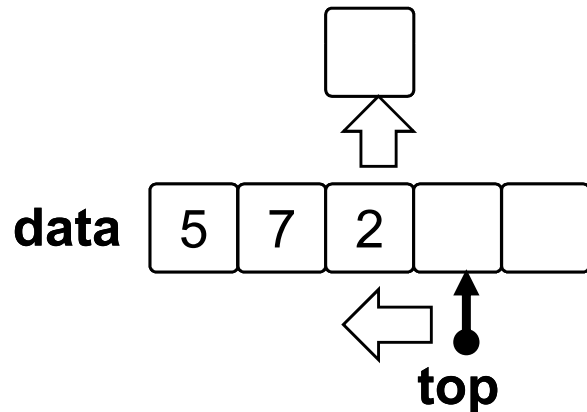


■ Cài đặt ngăn xếp trong C:

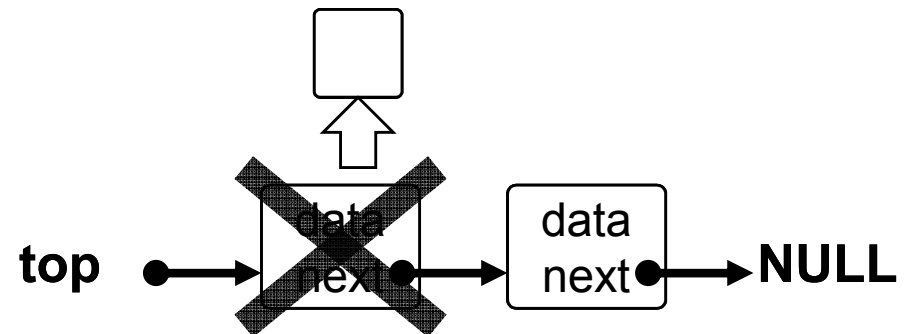
■ Các thao tác:

- Lấy phần tử: pop.

Dùng mảng động



Dùng danh sách liên kết đơn



Tổng quan về ngăn xếp

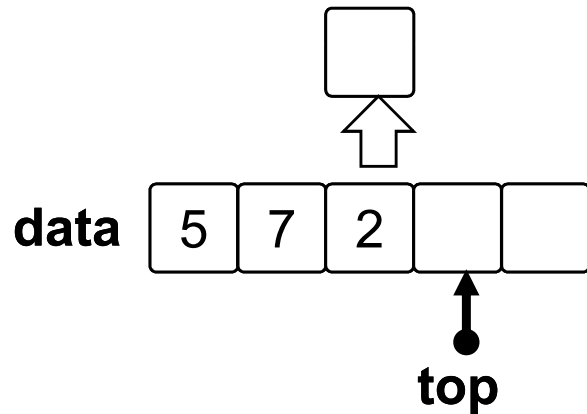


■ Cài đặt ngăn xếp trong C:

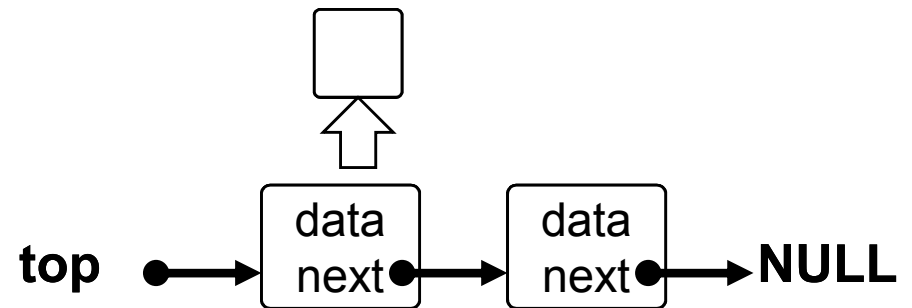
■ Các thao tác:

- Đọc phần tử: peek.

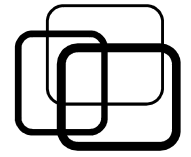
Dùng mảng động



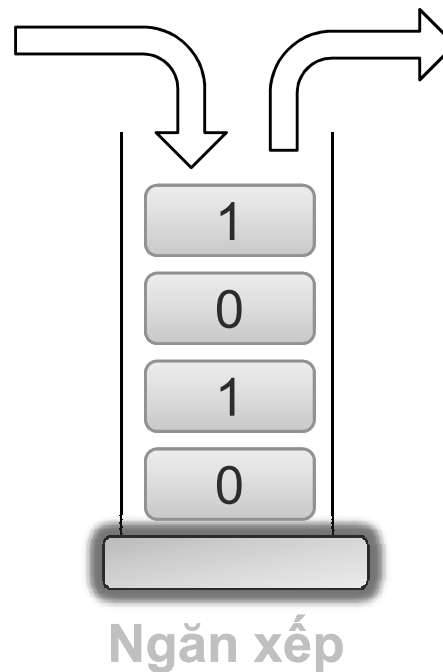
Dùng danh sách liên kết đơn



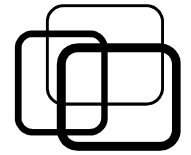
Tổng quan về ngăn xếp



- Ứng dụng ngăn xếp:
 - Thực hiện thao tác ngược:
 - Đổi 1 số nguyên sang hệ nhị phân.



Tổng quan về ngăn xếp



■ Ứng dụng ngăn xếp:

■ Cài đặt đệ quy:

- Dùng ngăn xếp hệ thống.
➔ Lỗi stack-overflow.
- Dùng ngăn xếp tự tạo
➔ Giải lập đệ quy.

```
long fibonacci(int n)
{
    stack_push(1);
    stack_push(1);

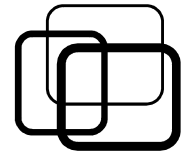
    for (int i = 2; i <= n; i++)
    {
        long fi1 = stack_pop();
        long fi2 = stack_pop();
        long fi = fi1 + fi2;
        stack_push(fi1);
        stack_push(fi);
    }

    return stack_pop();
}
```



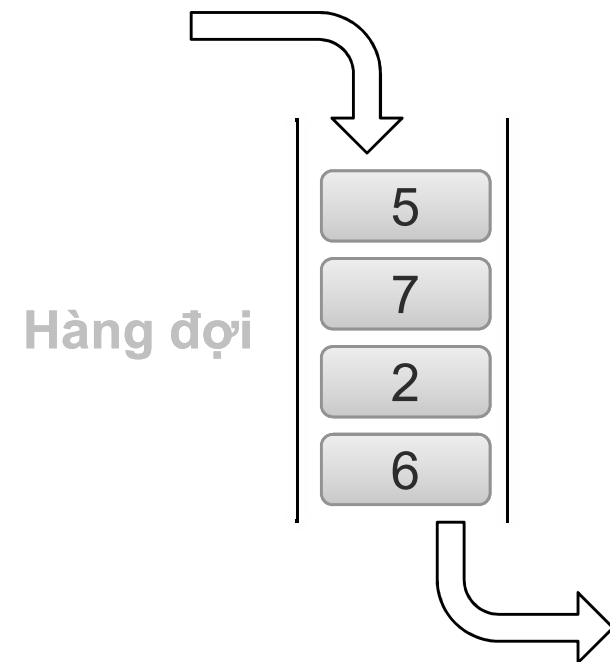
- Tổng quan về ngăn xếp.
- **Tổng quan về hàng đợi.**

Tổng quan về hàng đợi

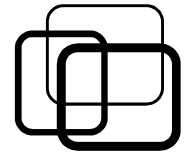


■ Khái niệm hàng đợi:

- Dãy phần tử hoạt động theo cơ chế FIFO.
- Cơ chế FIFO (**F**irst **I**n **F**irst **O**ut).
 - First Come First Serve.
 - Thứ tự vào trước ra trước.

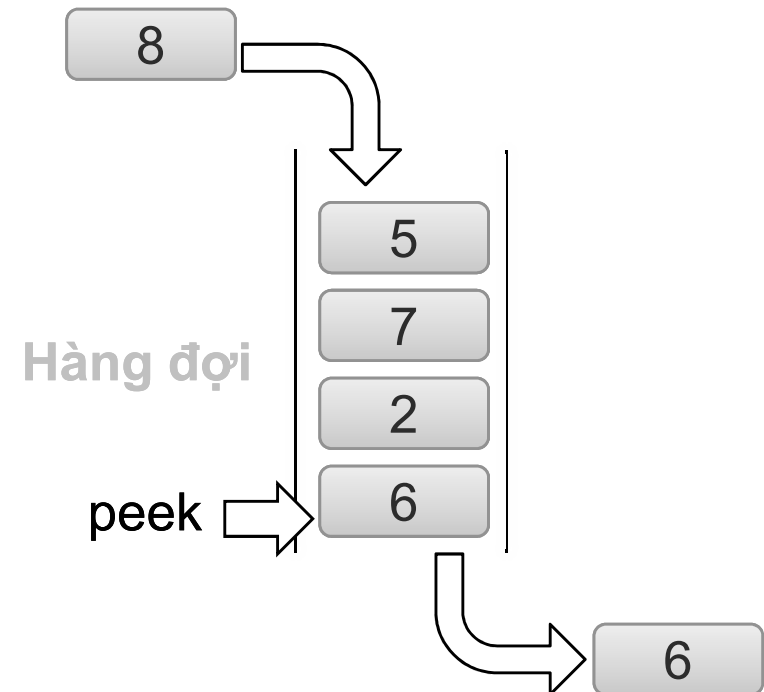


Tổng quan về hàng đợi



■ Các thao tác trên hàng đợi:

- push: thêm phần tử.
- pop: lấy phần tử.
- peek: đọc phần tử.
- init: khởi tạo.
- empty: kiểm tra rỗng.
- full: kiểm tra đầy.



Tổng quan về hàng đợi



■ Cài đặt hàng đợi trong C:

■ Khai báo và khởi tạo:

// Dùng mảng động.

```
struct Queue
```

```
{
```

```
    int *data;
```

```
    int  in;
```

```
    int  out;
```

```
    int  size;
```

```
};
```

// Dùng danh sách liên kết đơn.

```
struct Queue
```

```
{
```

```
    SNode *head;
```

```
    SNode *tail;
```

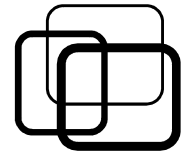
```
};
```


Tổng quan về hàng đợi



- Cài đặt hàng đợi trong C:
 - init.
 - isEmpty.
 - isFull.

Tổng quan về hàng đợi

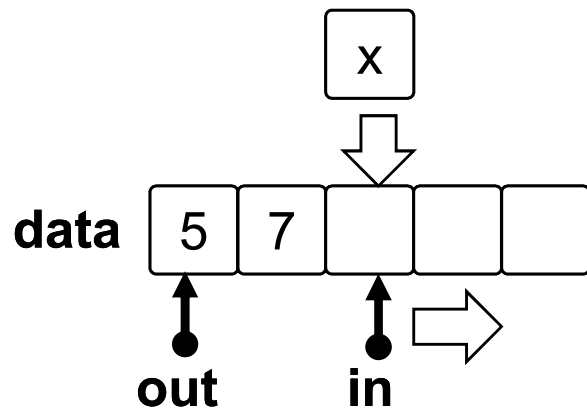


■ Cài đặt hàng đợi trong C:

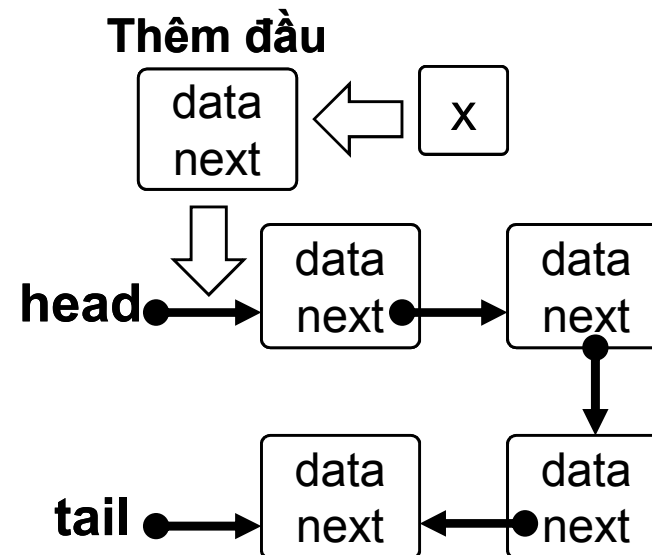
■ Các thao tác:

- Thêm phần tử: push.

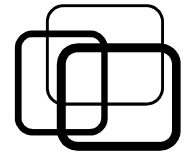
Dùng mảng động



Dùng danh sách liên kết đơn



Tổng quan về hàng đợi

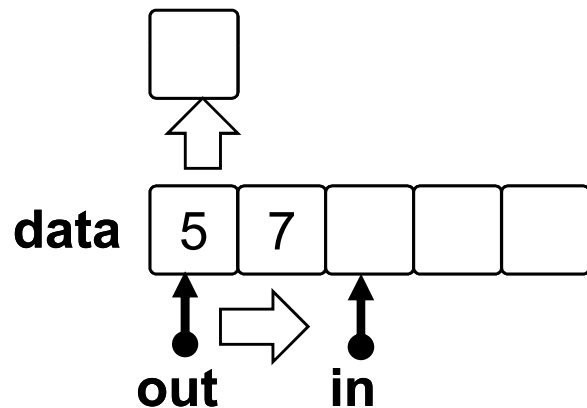


■ Cài đặt hàng đợi trong C:

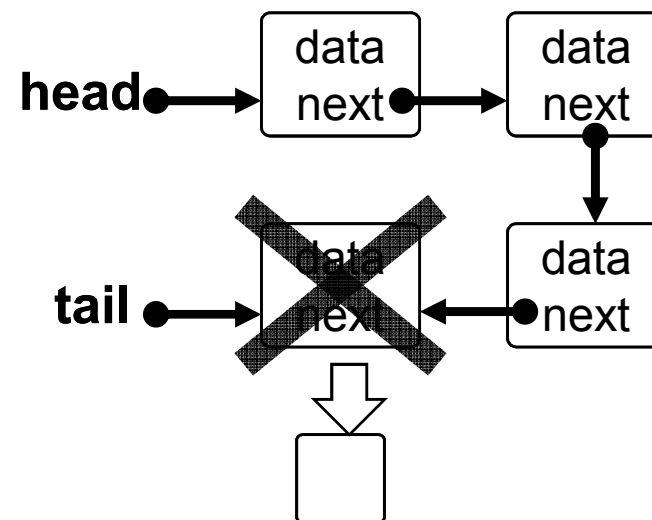
■ Các thao tác:

- Lấy phần tử: pop.

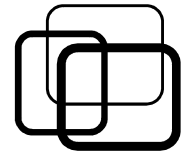
Dùng mảng động



Dùng danh sách liên kết đơn



Tổng quan về hàng đợi

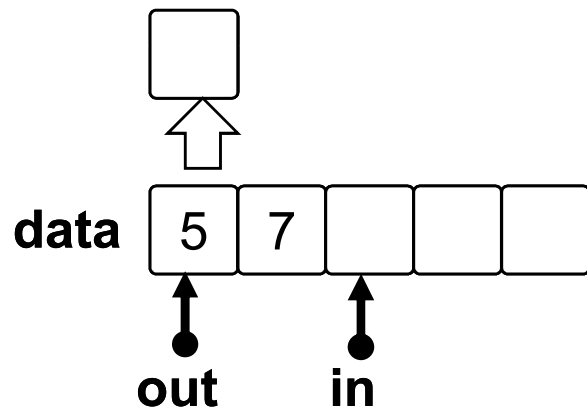


■ Cài đặt hàng đợi trong C:

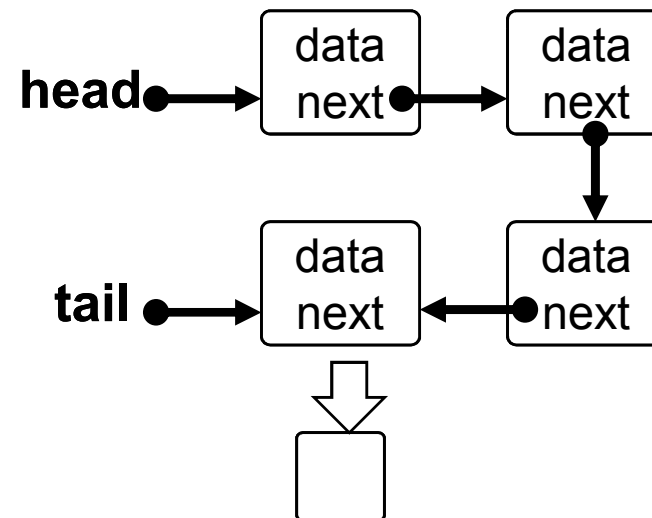
■ Các thao tác:

- Đọc phần tử: peek.

Dùng mảng động



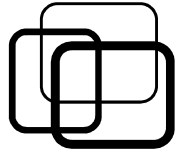
Dùng danh sách liên kết đơn



Tổng quan về hàng đợi



- Ứng dụng hàng đợi:
 - Hàng đợi hệ thống.
 - Duyệt cây theo chiều rộng.



■ Ngăn xếp:

- Dãy phần tử truy xuất LIFO.
- Phần tử vào sau ra trước.
- Các thao tác: push, pop, peek, empty, full.
- Cài đặt: mảng động, danh sách liên kết đơn.

■ Hàng đợi:

- Dãy phần tử truy xuất FIFO.
- Phần tử vào trước ra trước.
- Các thao tác: push, pop, peek, empty, full.
- Cài đặt: mảng động, danh sách vòng.

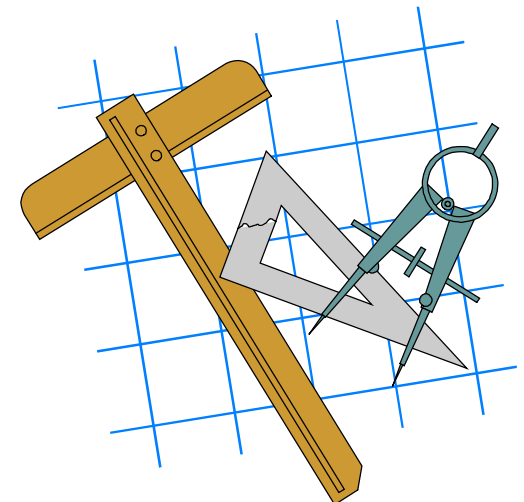


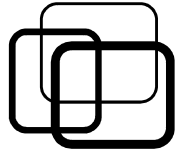


■ Bài tập 9.1:

Viết chương trình C (dùng ngăn xếp) để thực hiện:

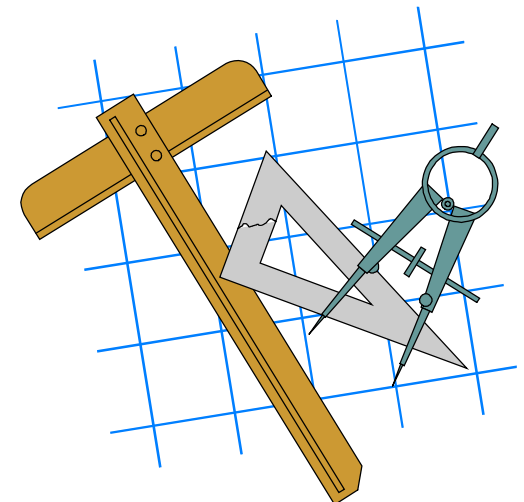
- Nhập một số nguyên N từ bàn phím.
- Xuất ra màn hình biểu diễn nhị phân của N.





■ Bài tập 9.2:

Cài đặt lại bài toán Tám Hậu bằng ngăn xếp (thay vì đệ quy).





■ Bài tập 9.3:

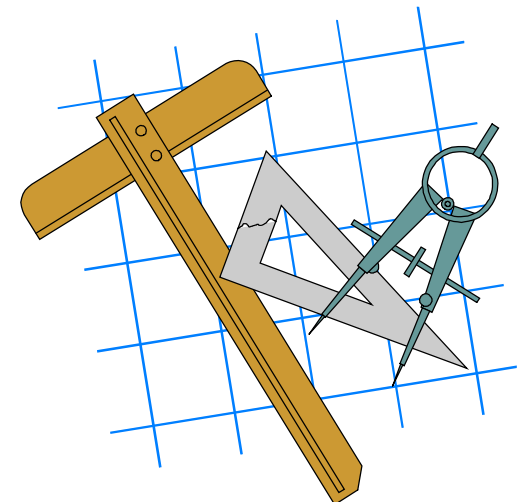
Viết chương trình C (dùng ngăn xếp) thực hiện việc sau:

- Nhập từ bàn phím một chuỗi biểu thức.

Ví dụ: $(2 * (4 + 3) / (1 - 5))$

- Tính giá trị biểu thức và xuất kết quả ra màn hình.

Thuật toán: xem slide kế...





■ Bài tập 9.3:

Thuật toán tính giá trị biểu thức infix:

- B1: Đọc 1 ký tự của chuỗi biểu thức.
- B2: Nếu là '(', số, phép toán thì push stack, quay lại B1.
Nếu là ')' thì
temp = pop stack.
Nếu temp là '(' thì quay lại B1.
số hạng 2 = temp.
toán tử = pop stack.
số hạng 1 = pop stack.
giá trị = số hạng 1 (toán tử) số hạng 2.
pop stack để loại dấu '('
push stack giá trị.
Quay lại B1.
- B3: kq = pop stack, xuất kq.

