



CẤU TRÚC DỮ LIỆU VÀ
GIẢI THUẬT

STACK &
QUEUE

BÀI HỌC HÔM NAY

- ▶ Stack
- ▶ Queue
- ▶ Cách sử dụng các thư viện STL có liên quan đến stack, queue

MINH HOẠ STACK

- ▶ LIFO: Last In First Out: Vào sau ra trước

pop()

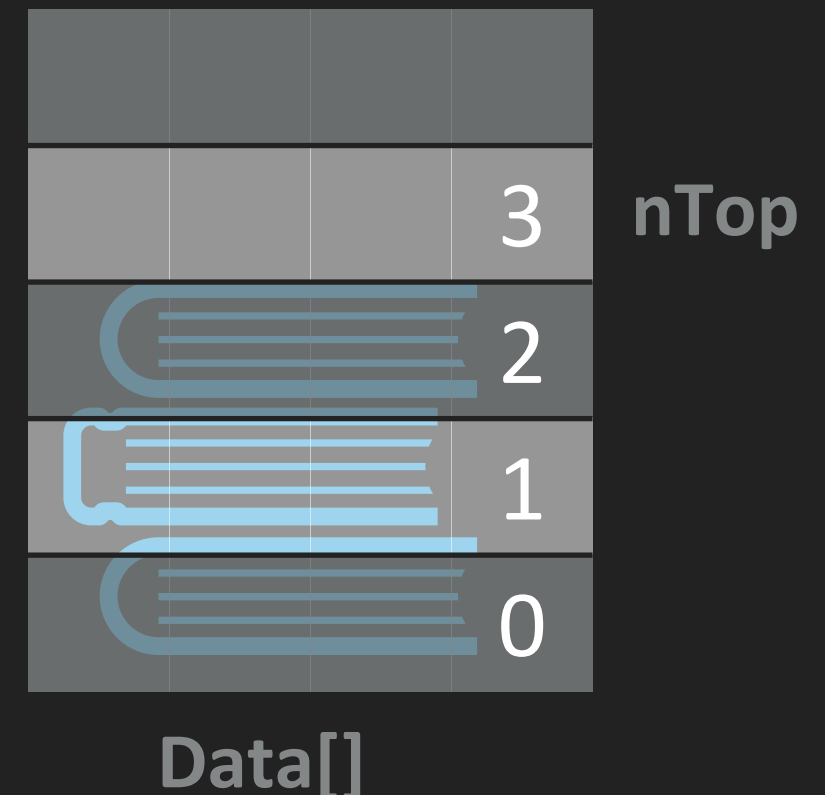
push()



CÀI ĐẶT STACK

- ▶ Khai báo 1 lớp Stack có các hàm sau:
 - ▶ isEmpty: kiểm tra ngăn xếp có rỗng không?
 - ▶ size: lấy kích thước của ngăn xếp
 - ▶ push: thêm 1 phần tử vào ngăn xếp
 - ▶ pop: Xóa 1 phần tử ra khỏi ngăn xếp
 - ▶ top: Chỉ đến phần tử đầu của ngăn xếp

```
class Stack
{
    int Data[MAX]
    int nTop
}
```



BÀI TẬP

- ▶ Cài đặt CTDL Stack



BÀI TẬP – CÀI ĐẶT LỚP STACK

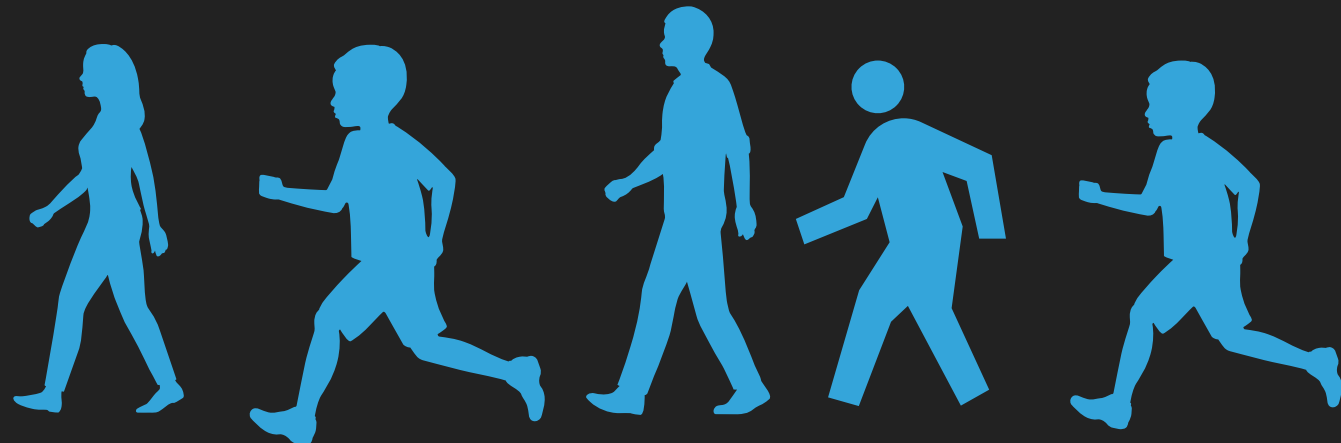
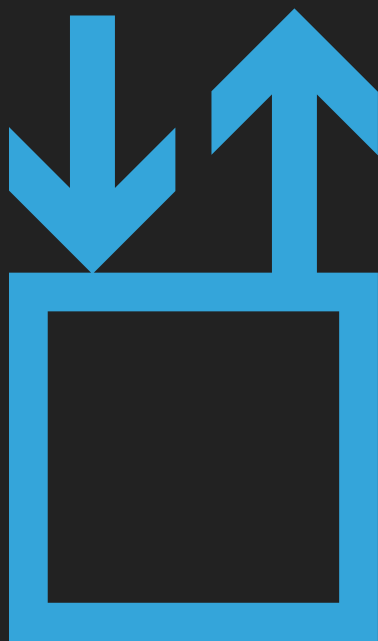
```
#include <iostream>
using namespace std;
#define MAX 30

class Stack
{
private:
    int Data[MAX];
public:
    int nTop;
    Stack()
    bool IsEmpty()
    void Push(int _data)
    int Pop()
    int Peek()
};

int main() {
    // insert code here...
    Stack q;
    q.push(1); //
    q.push(2); //
    q.push(3);
    cout<<q.pop()<<endl; // 3
    cout<<q.pop()<<endl; // 2
    q.push(4);
    cout<<q.pop()<<endl; // 4
    cout<<q.pop()<<endl; // 1
    return 0;
}
```

MINH HOẠ QUEUE

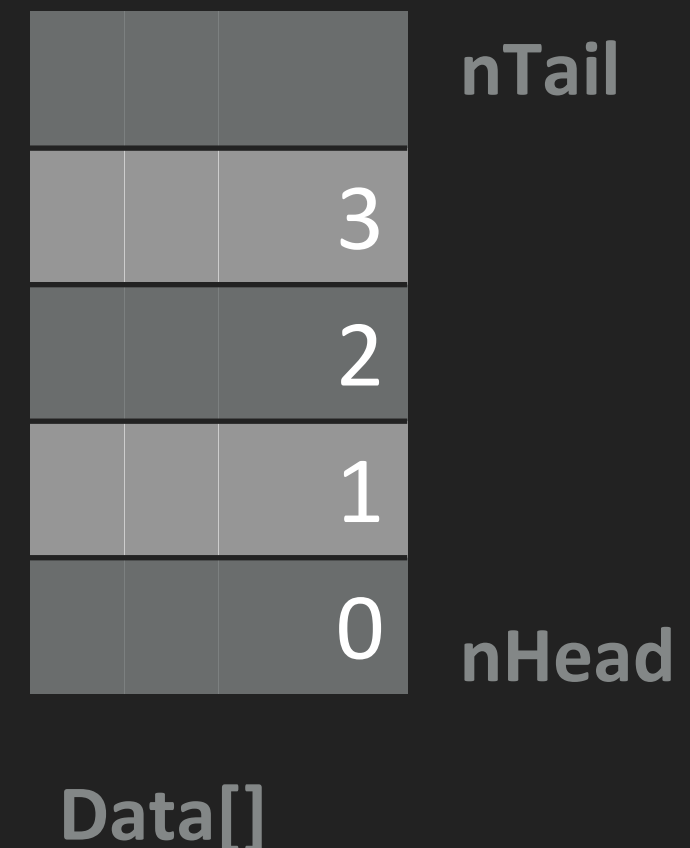
- ▶ FIFO: First In First Out: Vào trước ra trước



CÀI ĐẶT QUEUE

- ▶ Khai báo cấu trúc dữ liệu Queue có các hàm sau:
 - ▶ isEmpty: kiểm tra hàng đợi có rỗng không?
 - ▶ size: lấy kích thước của hàng đợi
 - ▶ push: thêm 1 phần tử vào hàng đợi
 - ▶ pop: xóa 1 phần tử ra khỏi hàng đợi
 - ▶ front: Chỉ đến phần tử đầu của hàng đợi

```
class Queue
{
    int Data[MAX]
    int nHead, nTail;
}
```



BÀI TẬP – CÀI ĐẶT LỚP QUEUE

```
#include <iostream>
using namespace std;
#define MAX 7

class Queue
{
private:
    int Data[MAX];
public:
    int nHead, nTail;
    Queue()
    bool IsEmpty()
    void Add(int _data)
    int Remove()
    int Peek()
};
```

```
int main() {
    // insert code here...
    Queue q;
    q.Add(1);
    q.Add(2);
    q.Add(3);
    cout<<q.Remove()<<q.IsEmpty()<<endl;
    q.Add(4);
    q.Add(5);
    cout<<q.Remove()<<q.IsEmpty()<<endl;
    q.Add(6);
    q.Add(7);
    cout<<q.Remove()<<q.IsEmpty()<<endl;
    cout<<q.Remove()<<q.IsEmpty()<<endl;
    return 0;
}
```

CÀI ĐẶT QUEUE



DEMO TRÊN LỚP

- ▶ Biểu diễn stack/queue bằng
 - ▶ Mảng tĩnh
 - ▶ Danh sách liên kết
- ▶ Sau đó bỏ class trên và thay bằng Standard Template Library (STL) mà không đổi hàm main

DÙNG STACK BẰNG STL THAY VÌ CLASS TỰ VIẾT

```
#include <iostream>
#include <stack>
using namespace std;
#define MAX 30
class Stack
int main() {
    // insert code here...
    stack<int> q;
    q.push(1); //
    q.push(2); //
    q.push(3);
    cout<<q.pop()<<endl; // 3
    cout<<q.pop()<<endl; // 2
    q.push(4);
    cout<<q.pop()<<endl; // 4
    cout<<q.pop()<<endl; // 1
    return 0;
}
```

BÀI TẬP

- ▶ Tính toán giá trị biểu thức
 - ▶ Tính toán giá trị $(5 + ((1 + 4) * (3 * 2)))$
 - ▶ Chuyển từ trung tố sang hậu tố
 - ▶ Trung tố: Infix
 - ▶ Tiền tố/Hậu tố: Prefix và Postfix
- ▶ Sử dụng Stack

BÀI TẬP

- ▶ Làm một bài tập trên trang HackerRank
- ▶ Tính giá trị biểu thức: $5 + ((1 + 4) * (3 * 2))$

KẾT THÚC

- ▶ Queue: Nên tập thói quen xếp hàng
- ▶ Stack: Ủh.... [StackOverFlow.com](https://stackoverflow.com)