

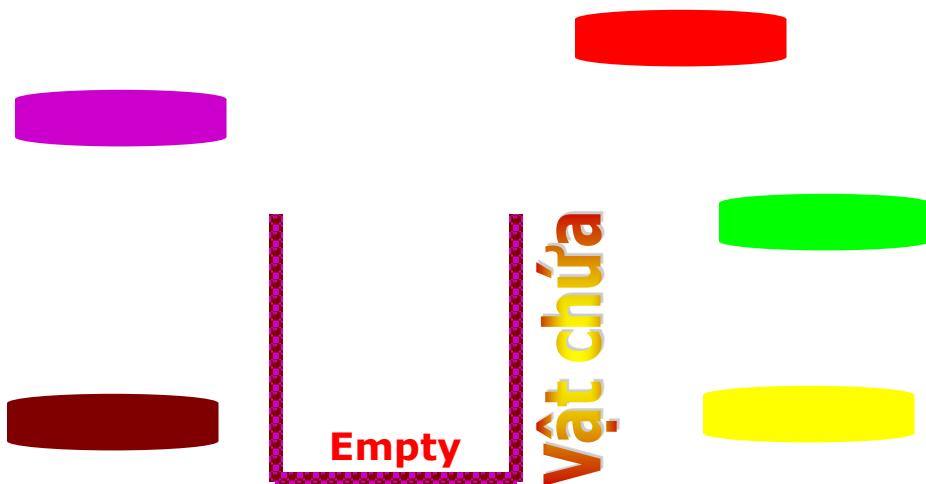
# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT



GV : ThS. Trần Văn Thọ  
E-mail : thotv@hufi.edu.vn

## VÍ DỤ DẪN NHẬP 1

❖ Quan sát việc xếp và lấy đĩa như sau:



## VÍ DỤ DẪN NHẬP 2

**Ví dụ:** Đổi số  $n = 25$  từ hệ thập phân sang hệ nhị phân

**Kết quả:**  $25_{10} =$  2

The diagram illustrates the conversion of the decimal number 25 to binary. It shows the process of dividing 25 by 2 repeatedly to find the quotient and remainder at each step. The remainders are highlighted in green, and the quotients are highlighted in blue.

Empty

Vật chứa

ThS. Trần Văn Thọ

3

**Môn: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**CHƯƠNG IV:**  
**STACK VÀ QUEUE**

# Stack

ThS. Trần Văn Thọ

5

## GIỚI THIỆU

- ❖ **Stack** là một đối tượng chứa các đối tượng khác, các đối tượng được **thêm vào** và **lấy ra** chỉ được thực hiện ở một đầu.
- ❖ Vì thế, việc thêm một đối tượng vào Stack hoặc lấy một đối tượng ra khỏi Stack được thực hiện theo cơ chế **LIFO** (**Last In First Out - Vào sau ra trước**).
- ❖ Các đối tượng có thể được thêm vào Stack bất kỳ lúc nào nhưng chỉ có **đối tượng thêm vào sau cùng** mới được phép **lấy ra khỏi Stack trước**.

ThS. Trần Văn Thọ

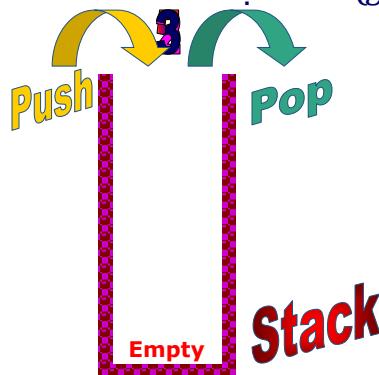
6

## GIỚI THIỆU

### ❖ Stack hỗ trợ 2 thao tác chính:

- **Push:** Thêm một đối tượng (*phần tử*) vào Stack.
- **Pop:** Xóa một đối tượng (*phần tử*) ra khỏi Stack.

→ cả 2 thao tác này chỉ diễn ra ở một đầu (*gọi là đỉnh*) của Stack.



ThS. Trần Văn Thọ

7

## GIỚI THIỆU

### ❖ Stack hỗ trợ các thao tác phụ khác:

- **IsEmpty:** Kiểm tra xem Stack có rỗng không.
- **GetTop:** Trả về giá trị của phần tử nằm ở đỉnh Stack mà không hủy nó ra khỏi Stack.

ThS. Trần Văn Thọ

8

## GIỚI THIỆU

❖ Stack thích hợp lưu trữ các loại dữ liệu mà trình tự truy xuất ngược với trình tự lưu trữ.

❖ Một số ứng dụng của Stack:

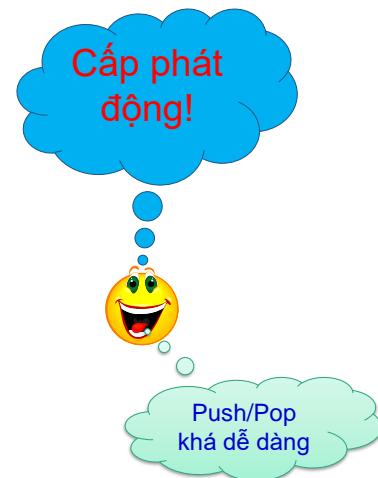
- Trong trình biên dịch/thông dịch, khi thực hiện Stack được sử dụng để lưu môi trường của các thủ tục.
- Tổ chức lưu vết các quá trình tìm kiếm theo chiều sâu và quay lui, vét cạn như: tìm đường đi ngắn nhất, bài toán 8 hậu, mã đi tuần,...
- Khử đệ qui: Tháp Hà Nội, QuickSort, ...
- Tính giá trị biểu thức toán học, đổi cơ số, đảo chuỗi,
- ...

## HỆN THỰC STACK

Mảng 1 chiều



Danh sách LK



## TỔ CHỨC STACK THEO D.SÁCH LIÊN KẾT

### Danh sách Liên kết

Cấp phát động!

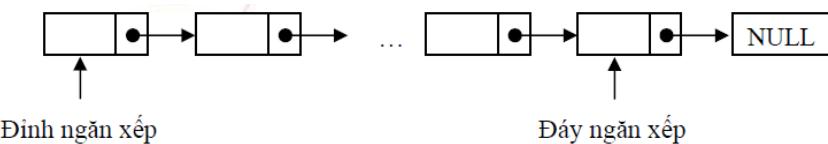


ThS. Trần Văn Thọ

11

### Hiện thực Stack dùng DSLK

- ❖ Có thể tạo một Stack bằng cách sử dụng một danh sách liên kết đơn (DSLK).
- ❖ Theo tính chất của danh sách liên kết đơn → phần tử nằm ở đầu danh sách là dễ thêm và xóa nhất.
- ❖ Do đó ta chọn:
  - Phần tử nằm ở đầu danh sách chính là đỉnh của ngăn xếp.
  - Phần tử nằm ở cuối danh sách là đáy ngăn xếp.



ThS. Trần Văn Thọ

12

## Hiện thực Stack dùng DSLK

❖ **Khai báo ngăn xếp bằng danh sách liên kết:**

- Chỉ cần 1 biến để lưu phần tử đầu tiên trong ngăn xếp (Top).

❖ **Khởi tạo ngăn xếp rỗng:**

- Biến Top được khởi tạo là NULL.

❖ **Kiểm tra ngăn xếp rỗng:**

- Ngăn xếp rỗng khi Top là NULL.

❖ **Kiểm tra ngăn xếp đầy:**

- Ngăn xếp không có giới hạn về kích thước do đó phương thức này không sử dụng.

## Hiện thực Stack dùng DSLK

❖ **Thêm một phần tử vào ngăn xếp:**

- Tạo ra phần tử mới.
- Cho biến Next của phần tử này trở tới Top hiện thời.
- Top bây giờ sẽ là phần tử mới này.

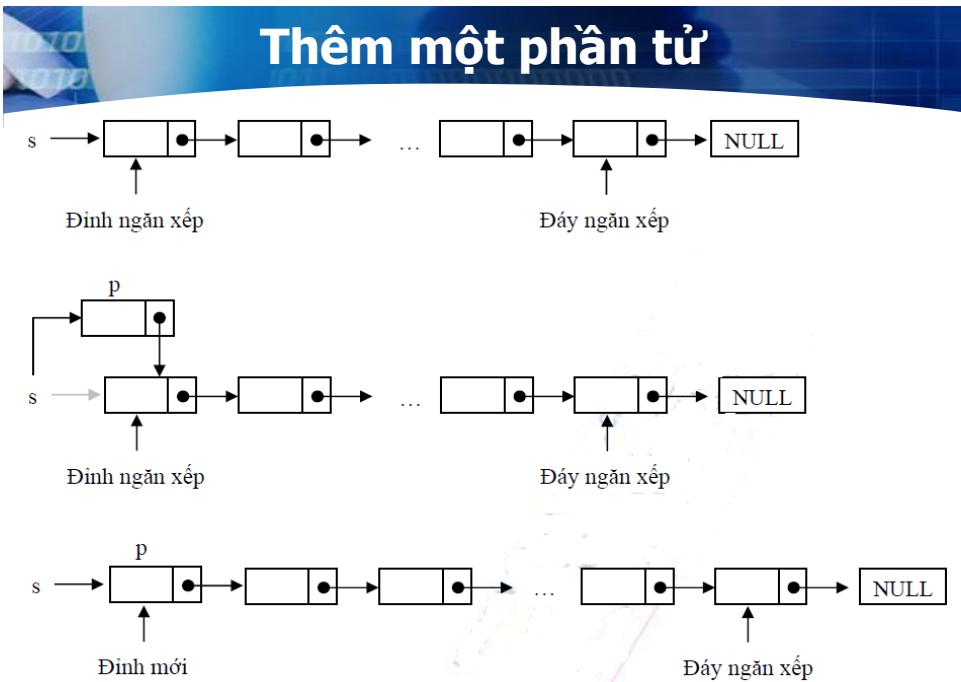
❖ **Lấy một phần tử trong ngăn xếp:**

- Phần tử được lấy chính là Top của ngăn xếp.

❖ **Xóa 1 phần tử khỏi ngăn xếp:**

- Top bây giờ là Top → Next.

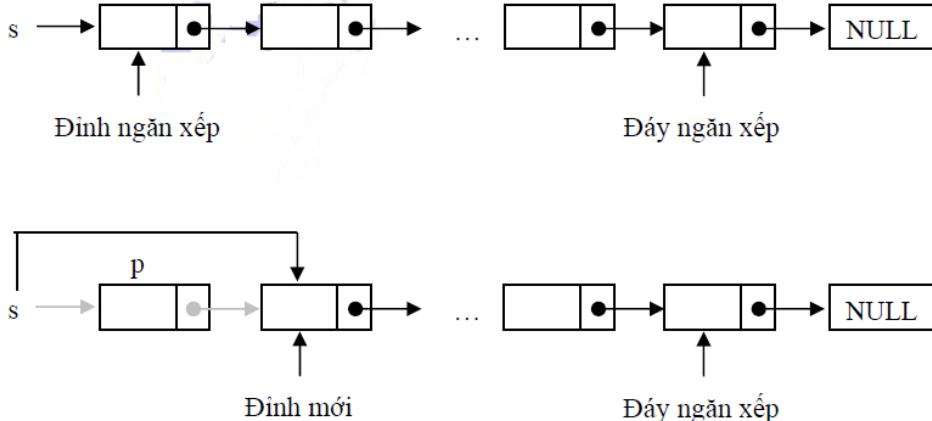
## Thêm một phần tử



ThS. Trần Văn Thọ

15

## Xóa một phần tử



ThS. Trần Văn Thọ

16

## KHAI BÁO STACK

Tạo cấu trúc dữ liệu cho Stack theo DSLK đơn:

```
typedef int ItemType;
struct StackNode
{
    ItemType Info;      //Thành phần dữ liệu của StackNode
    StackNode* Next;   //Con trỏ đến StackNode kế sau
};
struct Stack
{
    StackNode* Top;   //Con trỏ đến đỉnh của Stack
};
```

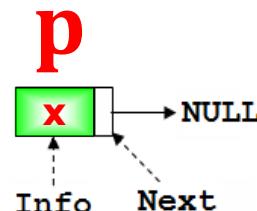
ThS. Trần Văn Thọ

17

## Tạo nút mới chứa giá trị x bất kỳ

❖ Hàm trả về địa chỉ phần tử mới tạo:

```
StackNode* createStackNode(ItemType x)
{
    StackNode* p = new StackNode;
    if(p == NULL) {
        printf("Không đủ bộ nhớ!");
        getch();
        return NULL;
    }
    p->Info = x;
    p->Next = NULL;
    return p;
}
```



ThS. Trần Văn Thọ

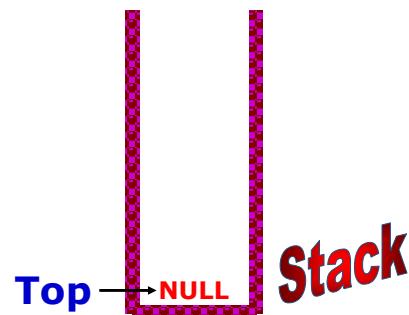
18

## CÁC THAO TÁC CƠ BẢN TRÊN STACK

- ❖ Stack là một cấu trúc dữ liệu trừu tượng tuyến tính có hai thao tác chính:
  - **push(Stack, O)**: Thêm đối tượng O vào Stack.
  - **pop(Stack, O)**: Nếu Stack không rỗng thì lấy đối tượng O ở đỉnh ra khỏi Stack và trả về giá trị của nó.
- ❖ Stack cũng có một số thao tác khác:
  - **initStack(Stack)**: Khởi tạo Stack rỗng.
  - **isEmpty(Stack)**: Kiểm tra Stack có rỗng không.

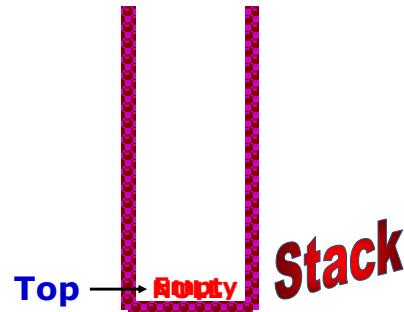
## KHỞI TẠO STACK

```
void initStack(Stack &s)
{
    s.Top = NULL;
}
```



## KIỂM TRA STACK RỖNG

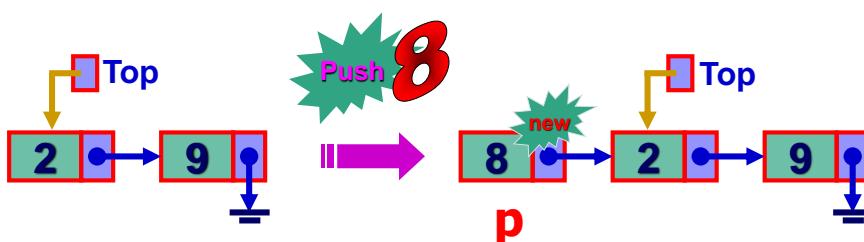
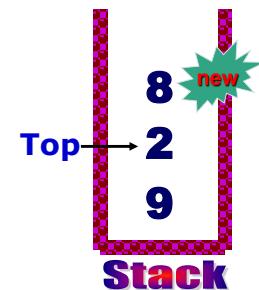
```
int isEmpty(Stack s)
{//1: Stack rỗng, 0: Stack không rỗng
    if(s.Top == NULL)
        return 1;
    else
        return 0;
}
```



## THÊM PHẦN TỬ VÀO STACK

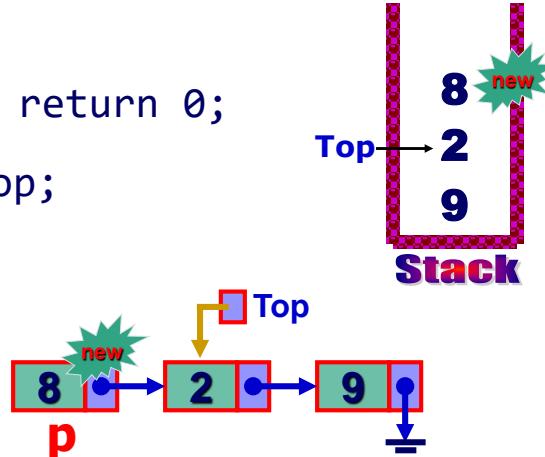
### ❖ Push

- Tạo nút mới
- Đưa nút mới vào đầu Stack



## THÊM PHẦN TỬ VÀO STACK

```
int push(stack &s, StackNode* p)
{
    if(p == NULL) return 0;
    p->Next = s.Top;
    s.Top = p;
    return 1;
}
```



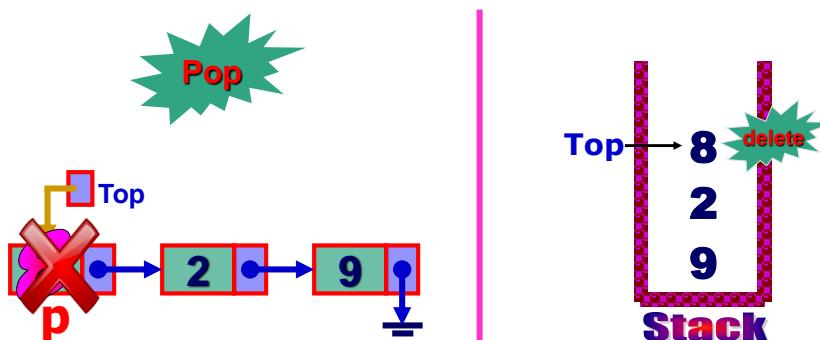
ThS. Trần Văn Thọ

23

## LẤY PHẦN TỬ RA KHỎI STACK

### ❖ Pop:

- Lấy phần tử đầu ra khỏi danh sách.
- Trả về nội dung và giải phóng nút.

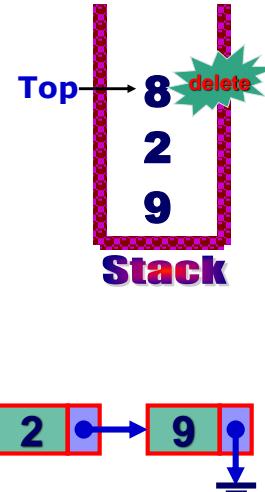


ThS. Trần Văn Thọ

24

## LẤY PHẦN TỬ RA KHỎI STACK

```
int pop(Stack &s, ItemType &x)
{
    if(isEmpty(s) == 1)
        return 0;
    StackNode* p = s.Top;
    s.Top = s.Top->Next;
    x = p->Info;
    delete p;
    return 1;
}
```



ThS. Trần Văn Thọ

25

## ỨNG DỤNG STACK

Stack có thể được ứng dụng để giải quyết một số bài toán như:

- Đổi cơ số, Đảo chuỗi ký tự, Tính giá trị biểu thức.
- Khử đệ quy: Bài toán tháp **HaNoi**, Thuật toán sắp xếp **QuickSort**, ...
- Áp dụng cho các bài toán dùng mô hình **LIFO** khác như: trình biên dịch/thông dịch, Tổ chức lưu vết các quá trình tìm kiếm theo chiều sâu và quay lui, vét cạn như: tìm đường đi ngắn nhất, bài toán 8 hậu, mã đi tuẫn, ...

ThS. Trần Văn Thọ

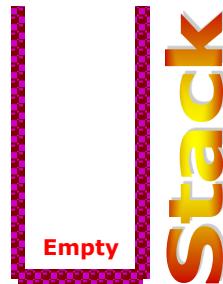
26

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

Ví dụ: Đổi số  $n = 25$  từ hệ thập phân sang hệ nhị phân

Kết quả:  $25_{10} =$

$$\begin{array}{r} 25 \\ \hline 2 | 12 + \text{dư } 1 \\ \hline 12 \\ \hline 2 | 6 + \text{dư } 0 \\ \hline 6 \\ \hline 2 | 3 + \text{dư } 0 \\ \hline 3 \\ \hline 2 | 1 + \text{dư } 1 \\ \hline 1 \\ \hline 2 | 0 + \text{dư } 1 \\ \hline \end{array}$$



ThS. Trần Văn Thọ

27

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

**Thuật toán:**

**Bước 1:** Khai báo Stack.

**Bước 2:** Khởi tạo Stack (*làm cho Stack rỗng*).

**Bước 3:** Kiểm tra trong khi n còn khác 0 thì lặp lại các công việc sau:

- Tính số dư của phép chia n cho a (*với a là cơ số*).
- Gán lại n bằng phần nguyên của phép chia n cho a.
- Bỏ số dư vừa tính được vào Stack.

**Bước 4:** Kiểm tra trong khi Stack còn chưa rỗng thì lần lượt lấy các giá trị ở đỉnh Stack ra sử dụng.

ThS. Trần Văn Thọ

28

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

29

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

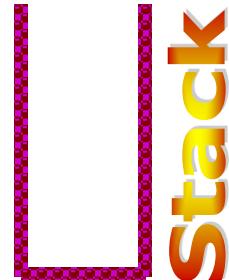
30

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

ThS. Trần Văn Thọ

31

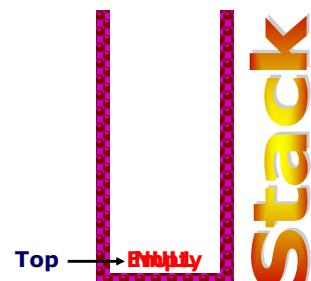


## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

ThS. Trần Văn Thọ

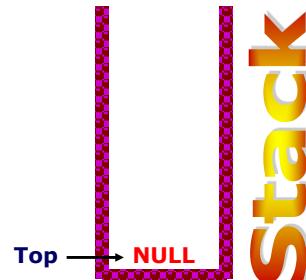
32



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

ThS. Trần Văn Thọ

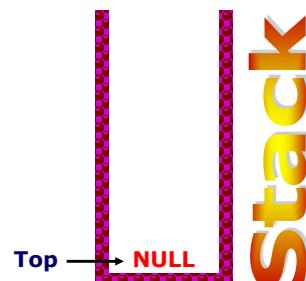


33

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

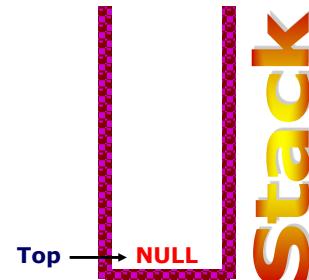
ThS. Trần Văn Thọ



34

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;           sodu = 1
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

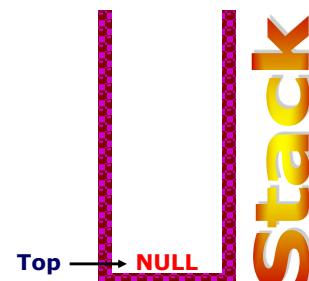


ThS. Trần Văn Thọ

35

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;           sodu = 1
        n /= a;               n = 12
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```



ThS. Trần Văn Thọ

36

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

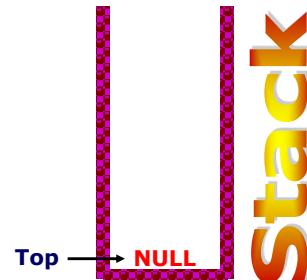
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 12
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

37



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

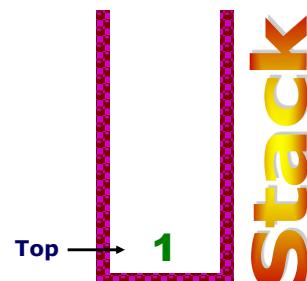
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 12
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

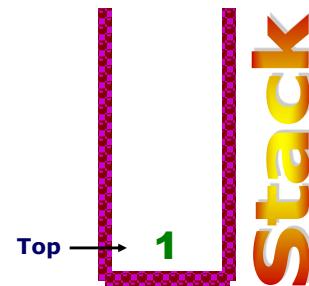
ThS. Trần Văn Thọ

38



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 12
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

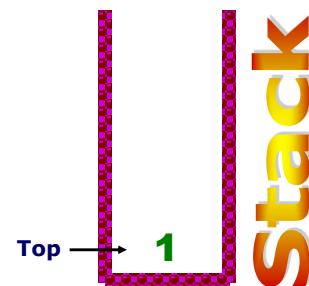


ThS. Trần Văn Thọ

39

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 12
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```



ThS. Trần Văn Thọ

40

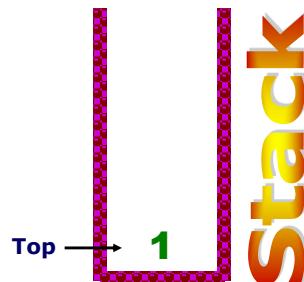
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 12
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



41

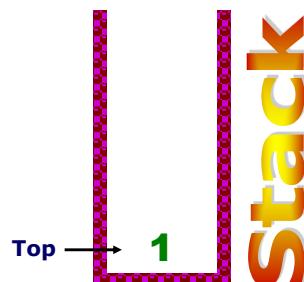
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 6
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



42

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

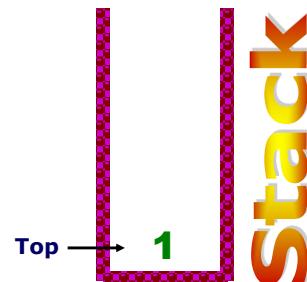
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 6
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

43



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

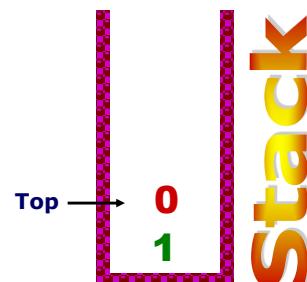
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 6
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

44



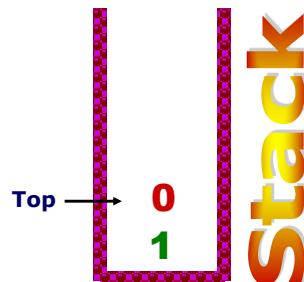
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 6
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



45

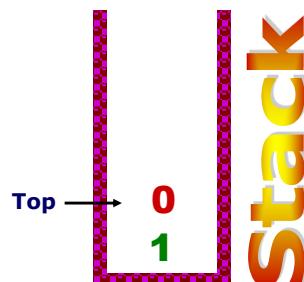
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 6
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



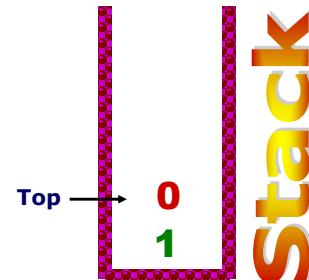
46

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 6
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

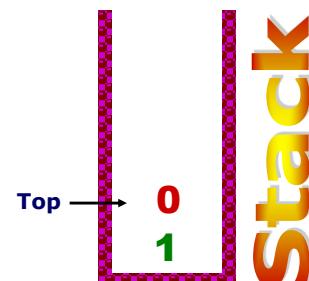
47

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 3
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

48

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 1
        n /= a;          n = 3
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

49



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

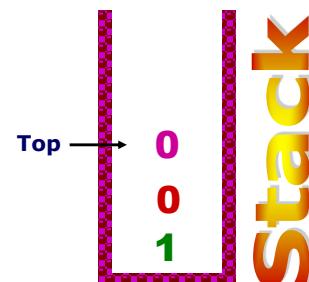
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 1
        n /= a;          n = 3
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

50



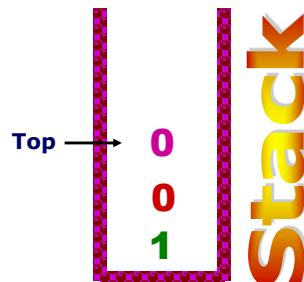
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 3
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



51

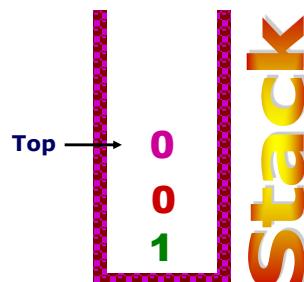
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 0
        n /= a;          n = 3
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

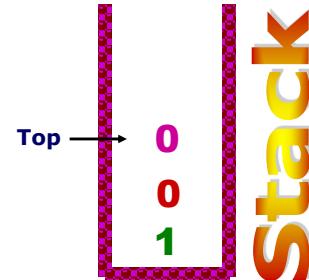
ThS. Trần Văn Thọ



52

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 3
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

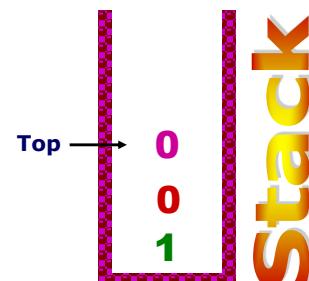


ThS. Trần Văn Thọ

53

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 1
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```



ThS. Trần Văn Thọ

54

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

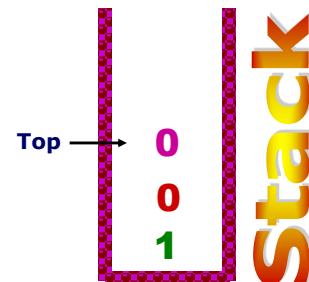
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 1
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

55



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

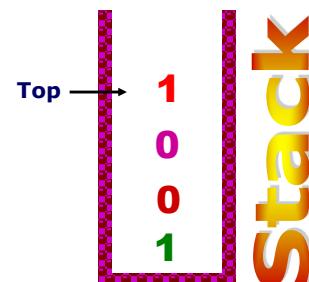
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 1
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

56

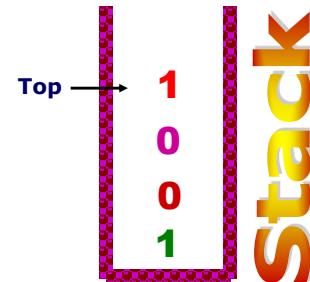


## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 1
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

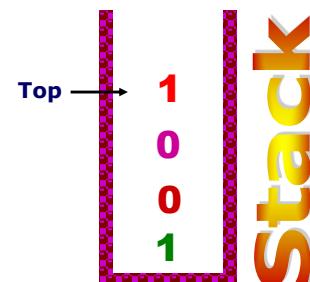
57

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 1
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

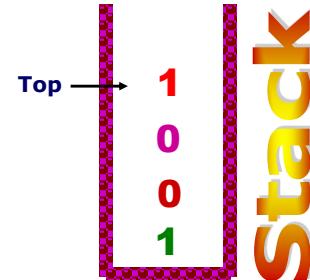
58

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 1
        n /= a;          n = 1
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

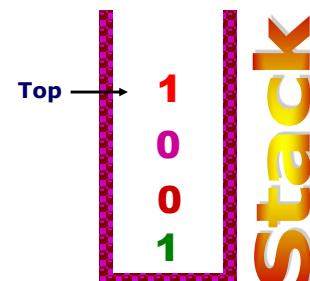
59

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;      sodu = 1
        n /= a;          n = 0
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

60

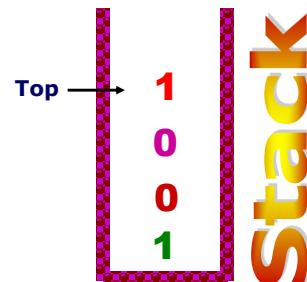
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 0
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



61

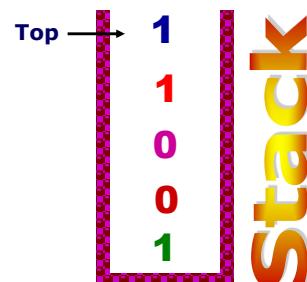
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 0
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



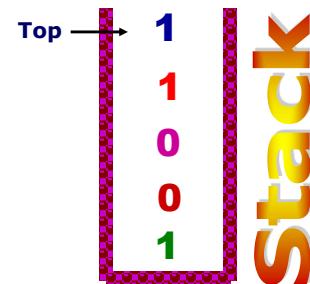
62

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 0
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

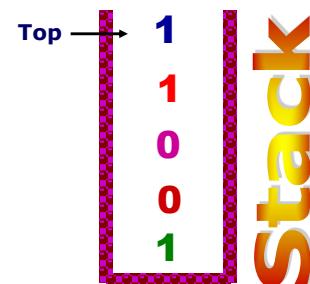
63

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;          sodu = 1
        n /= a;              n = 0
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}

```

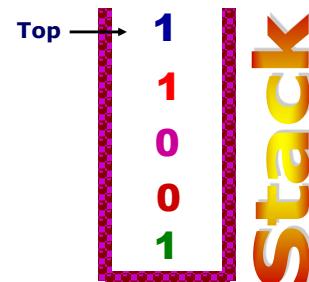


ThS. Trần Văn Thọ

64

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 =
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

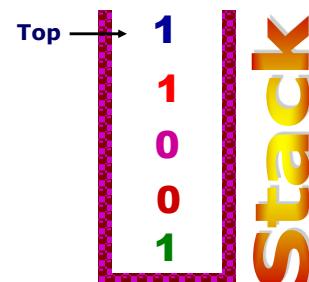


ThS. Trần Văn Thọ

65

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 =
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

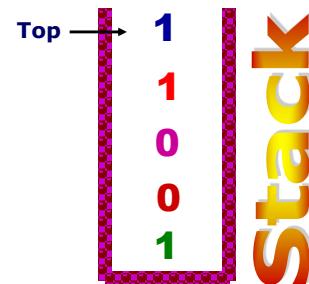


ThS. Trần Văn Thọ

66

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 =
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);
        printf("%3d", x);
    }
}
```

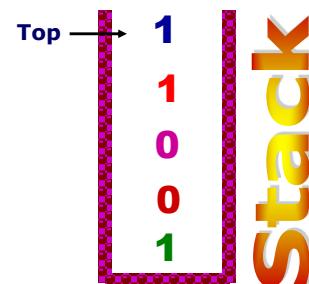


ThS. Trần Văn Thọ

67

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```
void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 =
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}
```



ThS. Trần Văn Thọ

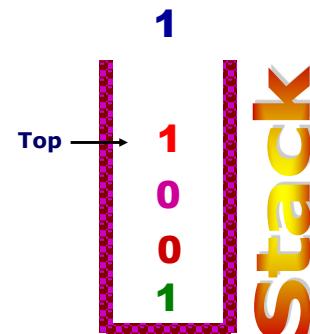
68

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 =
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

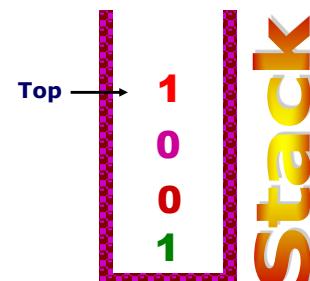
69

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

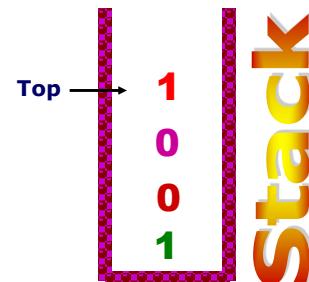
70

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

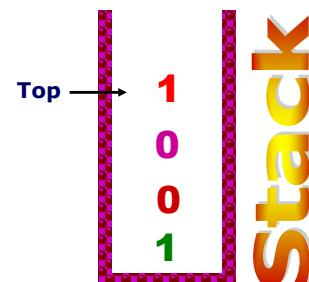
71

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

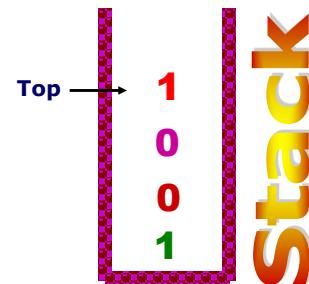
72

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

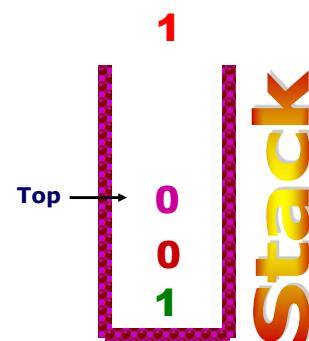
73

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

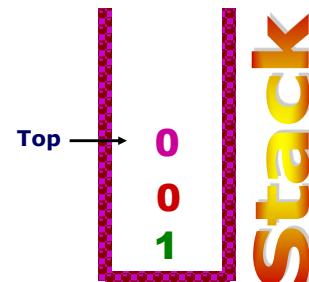
74

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

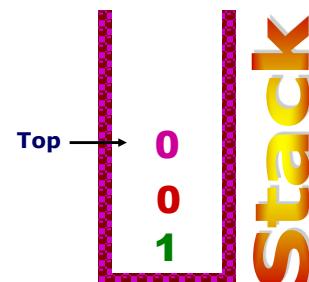
75

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

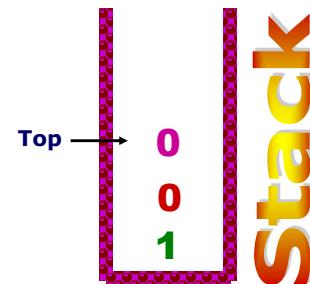
76

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

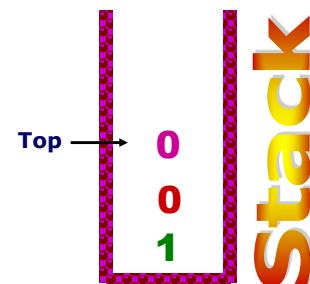
77

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

78

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

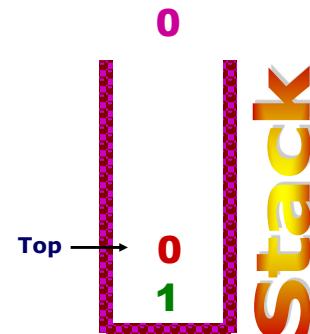
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

79



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

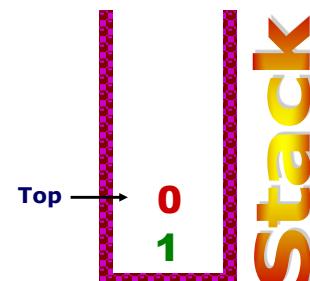
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

80

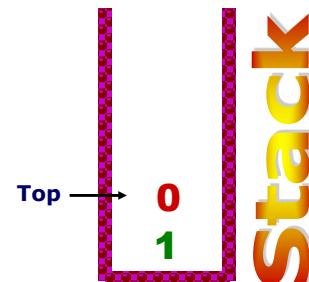


## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

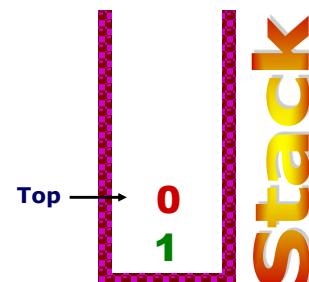
81

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

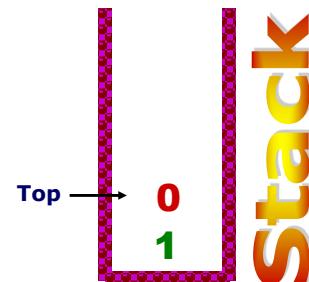
82

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

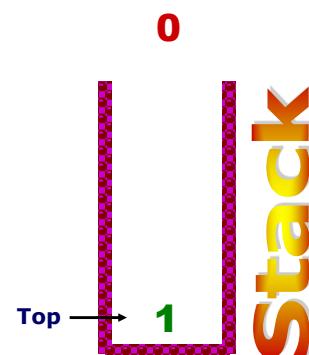
83

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

84

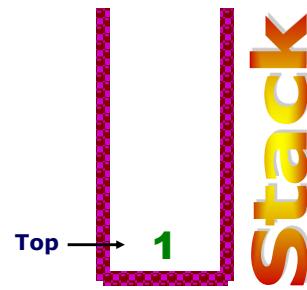
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



85

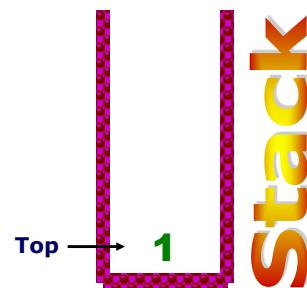
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



86

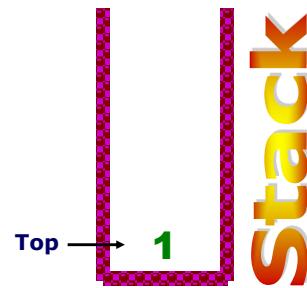
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 0
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



87

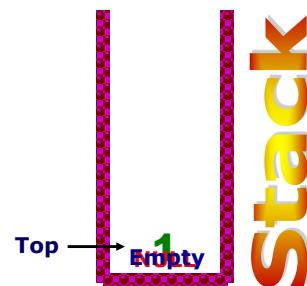
## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ



88

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

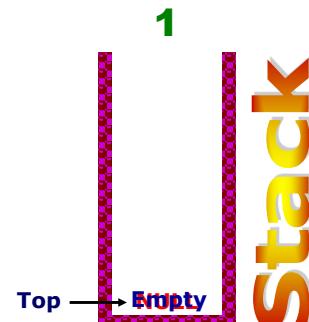
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0 0
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

89



## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

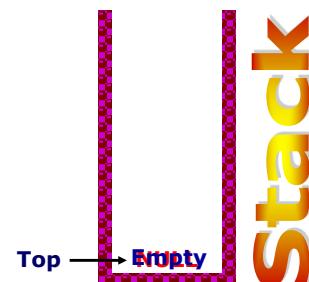
```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;
    initStack(stack);      Kết quả 2510 = 1 1 0 0 1
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```

ThS. Trần Văn Thọ

90

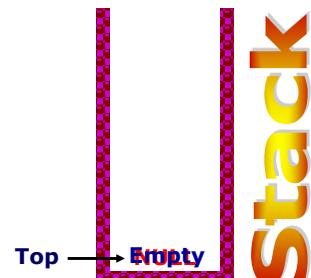


## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;          Kết quả 2510 = 1 1 0 0 1
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

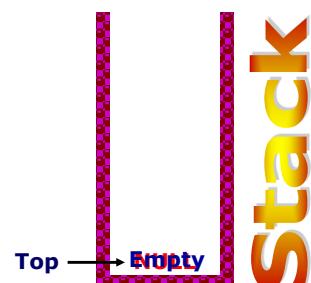
91

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;          Kết quả 2510 = 1 1 0 0 1
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

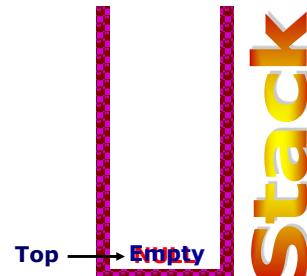
92

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;          Kết quả 2510 = 1 1 0 0 1
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      x = 1
        printf("%3d", x);
    }
}

```



ThS. Trần Văn Thọ

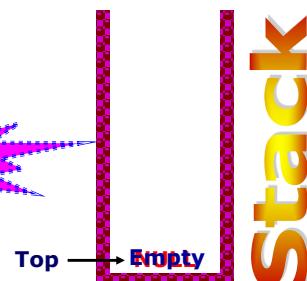
93

## ỨNG DỤNG STACK ĐỂ ĐỔI CƠ SỐ

```

void doiCoSo(int n, int a)    Giả sử n = 25, a = 2
{//Đổi từ hệ 10 sang một hệ số a bất kỳ.
    int sodu, x, cn = n;
    Stack stack;          Kết quả 2510 = 1 1 0 0 1
    initStack(stack);
    while(n != 0)
    {
        sodu = n%a;
        n /= a;
        push(stack, sodu);
    }
    printf("Kết quả %ld = ", cn);
    while(isEmpty(stack) == 0)
    {
        pop(stack, x);      Hoàn thành
        printf("%3d", x);
    }
}

```

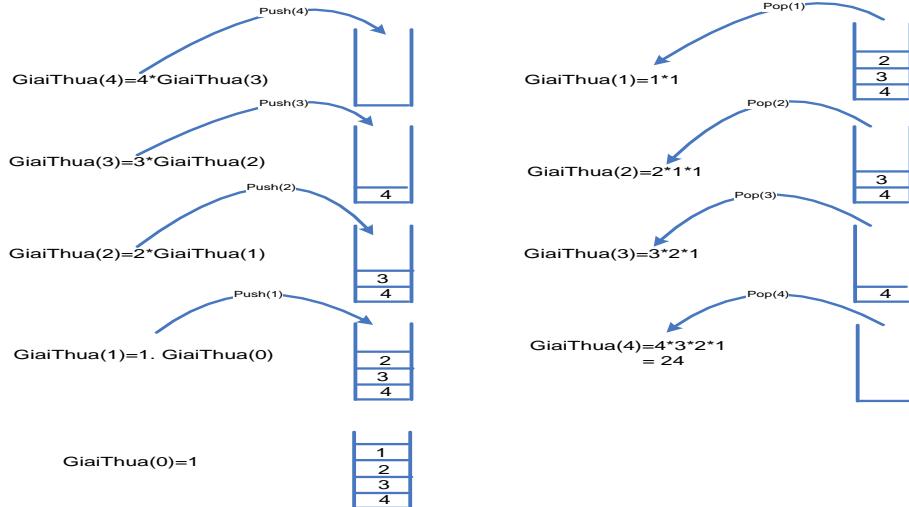


ThS. Trần Văn Thọ

94

# BÀI TẬP 1: TÍNH GIAI THỪA

❖ Dùng ngăn xếp khử đệ quy bài toán tính giai thừa (minh họa)



ThS. Trần Văn Thọ

95

# BÀI TẬP 1: TÍNH GIAI THỪA

```
long int tinhGiaiThua(int n) {
    DataTypeItem x;
    Stack stack;
    initStack(stack);
    while(n > 0)
        {//Thêm lần lượt nội dung vào stack
            StackNode* p = createStackNavigator(n);
            if(push(stack, p) == 0) break;
            n--;
        }
    long int kq = 1;
    while(1)
        {//Lấy lần lượt nội dung từ stack
            if(pop(stack, x) == 0) break;
            kq *= x;
        }
    return kq;
}
```

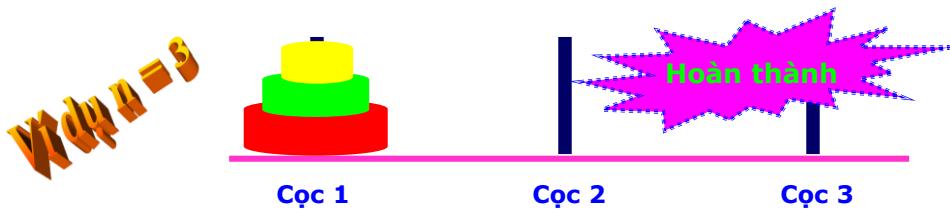
ThS. Trần Văn Thọ

96

## BÀI TẬP 2: THÁP HÀ NỘI

### ❖ Tháp Hà Nội:

- Có **n** đĩa đặt chồng lên nhau tại **cọc 1** sao cho đĩa lớn nằm dưới và đĩa nhỏ nằm trên.
- Chuyển **n** đĩa này sang **cọc 3**, cho phép lấy **cọc 2** làm trung gian, mỗi lần chỉ cho phép di chuyển 1 đĩa và vẫn giữ nguyên tắc đĩa lớn nằm dưới và đĩa nhỏ nằm trên.

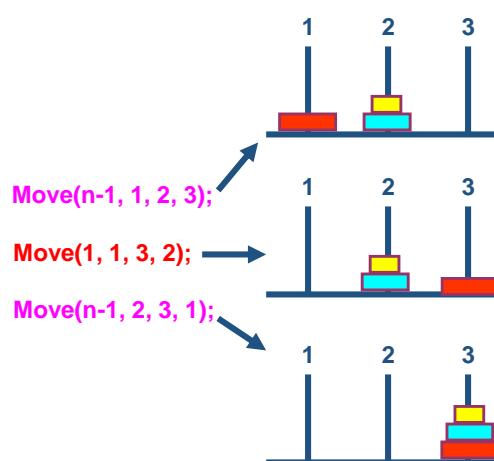
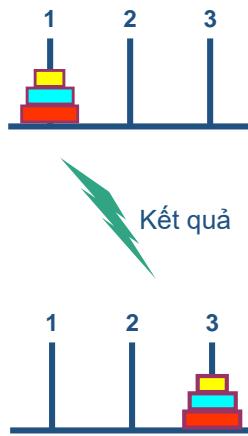


ThS. Trần Văn Thọ

97

## BÀI TẬP 2: THÁP HÀ NỘI

### ❖ Tháp HaNoi      Move( số đĩa, cọc nguồn, cọc đích, cọc tạm)



ThS. Trần Văn Thọ

98

## BÀI TẬP 2: THÁP HÀ NỘI

❖ Stack khử đệ quy:

- $\{n, src, des\}$
- Các cọc đánh số 1, 2, 3
- Cọc temp =  $6 - (src + des)$

1. Push  $\Rightarrow$  Stack: {n, 1, 3}



Stack lưu  
trữ thứ tự  
ngược

## BÀI TẬP 2: THÁP HÀ NỘI

2. Pop: Stack  $\Rightarrow \{n, src, des\}$

- Nếu  $n = 1$  thì: move src  $\Rightarrow$  des
- Ngược lại:
  - temp = **6** -  $(src + des)$
  - Push  $\Rightarrow$  Stack: {n - 1, temp, des}
  - Push  $\Rightarrow$  Stack: {1, src, des}
  - Push  $\Rightarrow$  Stack: {n - 1, src, temp}

## BÀI TẬP 3: THUẬT TOÁN QUICKSORT

**Ví dụ:** Giải thuật QuickSort dùng Stack để khử đệ quy:

- ❖ **Bước 1:** Left=0; Right=N-1;
- ❖ **Bước 2:** Chọn phần tử giữa  $x=a[(Left+Right) / 2]$
- ❖ **Bước 3:** Phân hoạch (Left, Right) thành (Left1, Right1) và (Left2, Right2) bằng cách xét:
  - y thuộc (Left1, Right1) nếu  $y \leq x$
  - y thuộc (Left2, Right2) ngược lại
- ❖ **Bước 4:** Nếu phân hoạch (Left2, Right2) có nhiều hơn 1 phần tử thì thực hiện:
  - Cất (Left2, Right2) vào Stack
  - Nếu (Left1, Right1) có nhiều hơn 1 phần tử thì thực hiện:
    - Left = Left1
    - Right = Right1
    - Quay lén bước 2
  - Ngược lại
    - Lấy (Left, Right) ra khỏi Stack, nếu Stack khác rỗng thì quay lén bước 2, ngược lại thì dừng

ThS. Trần Văn Thọ

101

## BÀI TẬP 3: THUẬT TOÁN QUICKSORT

**Cụ thể ý tưởng QuickSort khử đệ quy như sau:**

- **Push  $\Rightarrow$  Stack: {Left, Right}**
- Trong khi Stack còn chưa rỗng thì thực hiện:
  - **Pop: Stack  $\Rightarrow$  {Left, Right}**
  - Tính  $i = Left$ ,  $j = Right$ ,  $Mid = (Left + Right) / 2 \rightarrow$  phần tử chốt  $x$  là phần tử giữa.
  - **Thực hiện các công việc sau {**
    - Trong khi (phần tử thứ  $i <$  phần tử chốt) tăng  $i$  thêm 1.
    - Trong khi (phần tử thứ  $j >$  phần tử chốt) giảm  $j$  bớt 1.
    - Nếu ( $i \leq j$ ) { + Hoán vị 2 phần tử thứ  $i$  và thứ  $j$  với nhau.  
+ Tăng  $i$  thêm 1, đồng thời giảm  $j$  bớt 1 }
  - **} Lặp lại trong khi ( $i < j$ );**

ThS. Trần Văn Thọ

102

## BÀI TẬP 3: THUẬT TOÁN QUICKSORT

- Nếu ( $i < Right$ ) thì **Push**  $\Rightarrow$  Stack: { $i$ ,  $Right$ }.
- Nếu ( $j > Left$ ) thì **Push**  $\Rightarrow$  Stack: { $Left$ ,  $j$ }.
- Quá trình trên lặp lại cho đến khi **Stack** rỗng.

ThS. Trần Văn Thọ

103

## BÀI TẬP 3: THUẬT TOÁN QUICKSORT

**Bài tập:** Cài đặt thuật giải QuickSort không dùng đệ quy.

- Dùng DSLK, mỗi nút chứa thông tin biên trái và biên phải của đoạn chưa được sắp xếp.
- Áp dụng ý tưởng của slide trước để cài đặt.

ThS. Trần Văn Thọ

104

## BÀI TẬP 4: TÍNH GIÁ TRỊ BIỂU THỨC

### ❖ Thuật toán Ba Lan ngược (Reverse Polish Notation – RPN)

#### ▪ Định nghĩa RPN:

- Biểu thức toán học trong đó các toán tử được viết sau toán hạng và không dùng dấu ngoặc.
- Phát minh bởi Jan Lukasiewics một nhà khoa học Ba Lan vào những năm 1950.

ThS. Trần Văn Thọ

105

## BÀI TẬP 4: TÍNH GIÁ TRỊ BIỂU THỨC

**Infix** : toán tử viết giữa toán hạng

**Postfix (RPN)**: toán tử viết sau toán hạng

**Prefix** : toán tử viết trước toán hạng

Ví dụ:

**INFIX**

A + B

A \* B + C

A \* (B + C)

A - (B - (C - D))

A - B - C - D

**RPN (POSTFIX)**

A B +

A B \* C +

A B C + \*

A B C D - - -

A B C - D -

**PREFIX**

+ A B

+ \* A B C

\* A + B C

- A - B - C D

- - - A B C D

ThS. Trần Văn Thọ

106

## BÀI TẬP 4: TÍNH GIÁ TRỊ BIỂU THỨC

### Kỹ thuật gạch dưới:

1. Duyệt từ trái sang phải của biểu thức cho đến khi gặp toán tử.
2. Gạch dưới 2 toán hạng ngay trước toán tử và kết hợp chúng bằng toán tử trên.
3. Lặp đi lặp lại cho đến hết biểu thức.

**Ví dụ:**  $2 * ((3 + 4) - (5 - 6))$

$$\rightarrow 2 \underline{3} \underline{4} + 5 6 \dots *$$

$$\begin{aligned} &\rightarrow 2 \underline{\underline{3}} \underline{\underline{4}} + 5 6 \dots * \\ &\rightarrow 2 \underline{\underline{7}} \underline{\underline{5}} \underline{\underline{6}} \dots * \\ &\rightarrow 2 \underline{\underline{7}} \underline{\underline{5}} \underline{\underline{6}} - \dots * \\ &\rightarrow 2 \underline{\underline{7}} \underline{\underline{-1}} \dots * \\ &\rightarrow 2 \underline{\underline{7}} \underline{\underline{-1}} \dots * \rightarrow 2 \underline{\underline{8}} * \rightarrow \underline{\underline{2}} \underline{\underline{8}} * \rightarrow 16 \end{aligned}$$

ThS. Trần Văn Thọ

107

## Chuyển infix thành postfix

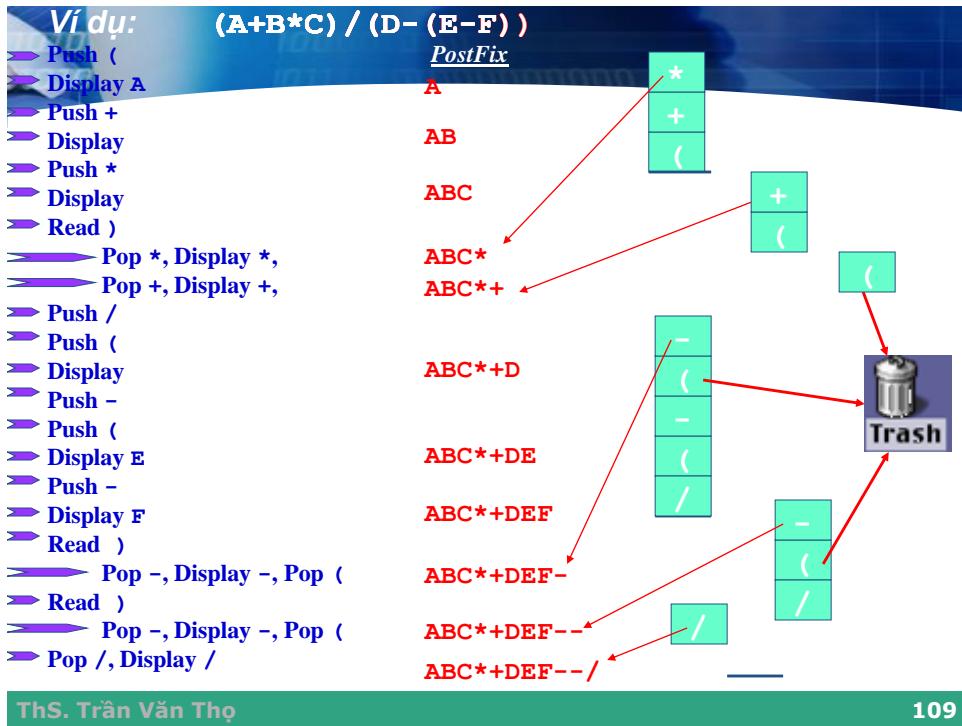
- ❖ 1. Khởi tạo Stack rỗng (*chứa các phép toán*).
- ❖ 2. Lặp cho đến khi kết thúc biểu thức:
  - Đọc 01 phần tử của biểu thức (*01 phần tử có thể là hằng, biến, phép toán, ')' hay '('*).
  - Nếu phần tử là:
    - 2.1 '**(**' : đưa vào Stack.
    - 2.2 '**)**' : lấy các phần tử của Stack ra cho đến khi lấy xong '**(**'.
    - 2.3 Một phép toán: **^ + - \* /**
      - Nếu **Stack rỗng**: đưa vào Stack.
      - Nếu Stack khác rỗng và **phép toán có độ ưu tiên cao hơn phần tử ở đầu Stack**: đưa vào Stack.
      - Nếu Stack khác rỗng và **phép toán có độ ưu tiên thấp hơn hoặc bằng phần tử ở đầu Stack**:
        - Lấy phần tử ở đỉnh Stack ra;
        - Sau đó lặp lại việc so sánh với phần tử ở đỉnh Stack.
    - 2.4 **Hằng hoặc biến**: đưa vào postfix.
  - ❖ 3. Lấy hết tất cả các phần tử của Stack ra.

**Độ ưu tiên:**

<b>^</b>	<b>cao nhất</b>
<b>*, /</b>	<b>cao nhì</b>
<b>+, -</b>	<b>cao ba</b>

ThS. Trần Văn Thọ

108



ThS. Trần Văn Thọ

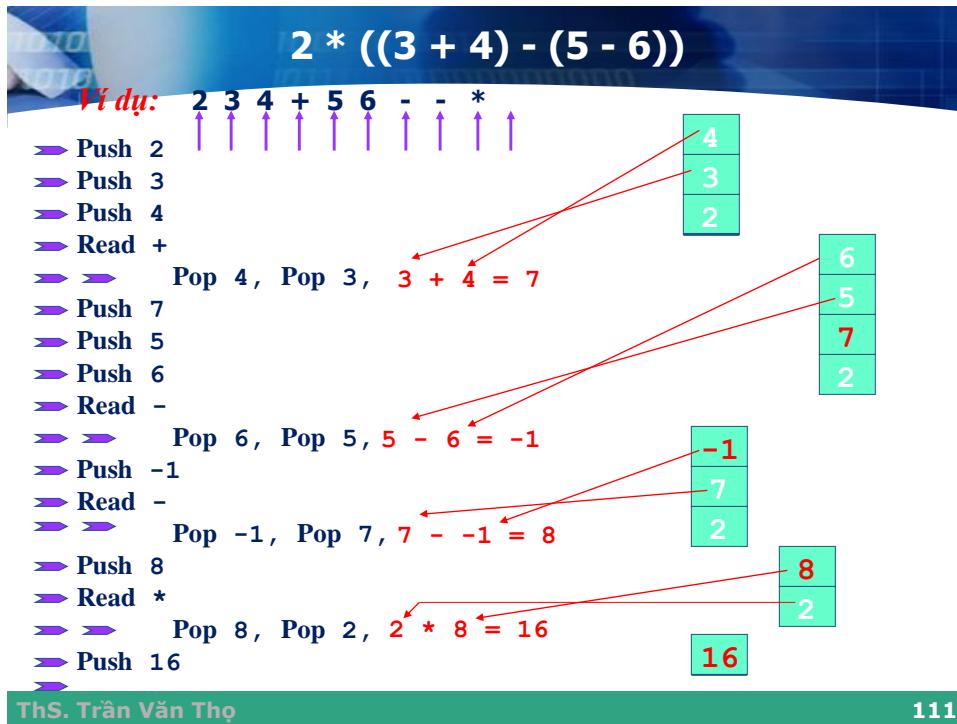
109

## BÀI TẬP 4: TÍNH GIÁ TRỊ BIỂU THỨC

- Khởi tạo Stack rỗng (*chứa hằng hoặc biến*).
- Lặp cho đến khi kết thúc biểu thức:
  - Đọc 01 phần tử của biểu thức (*hằng, biến, phép toán*).
  - Nếu phần tử là hằng hay biến: đưa vào Stack.
  - Ngược lại:
    - Lấy ra 2 phần tử của Stack ra.
    - Áp dụng phép toán cho 2 phần tử vừa lấy ra.
    - Đưa kết quả vừa tính được vào lại Stack.
- Giá trị của biểu thức chính là phần tử cuối cùng của Stack.

ThS. Trần Văn Thọ

110



## TÍNH GIÁ TRỊ BIỂU THỨC

❖ Giai đoạn 1: Chuyển biểu thức trung tố sang hậu tố

Trung tố (infix)		Hậu tố (Postfix)
$(6 / 2 + 3) * (7 - 4)$	↔	$6 2 / 3 + 7 4 - *$

Biểu thức hậu tố  
dễ tính toán hơn  
trong máy tính

ThS. Trần Văn Thọ      112

## TÍNH GIÁ TRỊ BIỂU THỨC

Chuyển từ trung tố về hậu tố.

❖ Duyệt qua từng phần tử trong infix (C)

- Nếu C là toán hạng thì bỏ vào postfix
- Nếu C là '(' thì push  $\Rightarrow$  Stack.
- Nếu C là ')' thì lấy tất cả phần tử trong Stack ra bỏ vào postfix cho đến khi gặp '('.

ThS. Trần Văn Thọ

113

## TÍNH GIÁ TRỊ BIỂU THỨC

▪ Nếu C là toán tử:

- Lấy các toán tử có độ ưu tiên cao hơn bỏ vào postfix.
- Bỏ toán tử vào Stack.

Ví dụ: **(2 \* 3 + 8 / 2) \* (5 - 1)**

ThS. Trần Văn Thọ

114

## TÍNH GIÁ TRỊ BIỂU THỨC

**(2 \* 3 + 8 / 2) \* (5 - 1)**

Đọc	Xử lý	Stack	postfix
(	Đẩy vào Stack	(	
2	Bỏ vào postfix	(	2
*	Do ở đỉnh Stack chứa '(' nên bỏ dấu '*' vào Stack.	(*	2
3	Bỏ vào postfix	( *	2 3
+	Do '+' có độ ưu tiên thấp hơn '*' ở đỉnh Stack nên ta lấy '*' ra và bỏ vào postfix. Do ở đỉnh Stack chứa '(' nên bỏ dấu '+' vào Stack.	( +	2 3 *
8	Bỏ vào postfix	( +	2 3 * 8
/	Do '/' có độ ưu tiên cao hơn '+' ở đỉnh Stack nên đưa '/' vào Stack.	( + /	2 3 * 8
2	BỎ VÀO POSTFIX	( + /	2 3 * 8 2

ThS. Trần Văn Thọ

115

## TÍNH GIÁ TRỊ BIỂU THỨC

Đọc	Xử lý	Stack	postfix
)	Lấy trong Stack ra cho đến khi gặp ngoặc '('		2 3 * 8 2 / +
*	Đưa vào Stack	*	2 3 * 8 2 / +
(	Đưa vào Stack	* (	2 3 * 8 2 / +
5	BỎ VÀO POSTFIX	* (	2 3 * 8 2 / + 5
-	Do trên đỉnh Stack chứa '(' nên bỏ dấu '-' vào Stack.	* ( -	2 3 * 8 2 / + 5
1	BỎ VÀO POSTFIX	* ( -	2 3 * 8 2 / + 5 1
)	Lấy trong Stack ra cho đến khi gặp ngoặc mở '('	*	2 3 * 8 2 / + 5 1 -
	Lấy những phần tử còn lại trong Stack và bỏ vào Stack.		2 3 * 8 2 / + 5 1 - *

ThS. Trần Văn Thọ

116

## TÍNH GIÁ TRỊ BIỂU THỨC

### ❖ Giai đoạn 2: Tính giá trị biểu thức postfix



Jan Lukasiewicz

Postfix không cần có dấu ngoặc vẫn có thể tính đúng bằng cách đọc lần lượt biểu thức từ trái qua phải và dùng một Stack để lưu trữ kết quả trung gian.

ThS. Trần Văn Thọ

117

## TÍNH GIÁ TRỊ BIỂU THỨC

### Ý tưởng:

- ❖ Khởi tạo Stack = { $\emptyset$ }
- ❖ Đọc lần lượt các phần tử từ trái, kiểm tra
  - Nếu toán hạng: Push  $\Rightarrow$  Stack
  - Nếu toán tử: lấy hai toán hạng, thực hiện phép toán, kết quả Push vào Stack
- ❖ Sau khi đọc xong, trong Stack còn duy nhất một phần tử  $\Rightarrow$  kết quả!

ThS. Trần Văn Thọ

118

## TÍNH GIÁ TRỊ BIỂU THỨC

$(2 * 3 + 8 / 2) * (5 - 1)$        $2 \ 3 * 8 \ 2 / + 5 \ 1 - *$

Đọc	Xử lý	Stack	Output
2,3	Đưa vào Stack	2, 3	
*	$2 * 3$	6	
8	Đưa vào Stack	6, 8	
2	Đưa vào Stack	6, 8, 2	
/	Lấy 8/2	6, 4	
+	Lấy 6 + 4	10	
5	Đưa vào Stack	10, 5	
1	Đưa vào Stack	10, 5, 1	
-	Lấy 5 - 1	10, 4	
*	Lấy 10 * 4	40	<b>40</b>

ThS. Trần Văn Thọ

119

## BÀI TẬP

$$3 * (2 + 6 * 2 / 3 - 1) - 2 * 3 / 2 + 1$$

ThS. Trần Văn Thọ

120

**Thank for your attention!**



**See you next week!**

**THAM KHẢO THÊM**

**TỔ CHỨC STACK THEO MÀNG 1 CHIỀU**

## TỔ CHỨC THEO MẢNG 1 CHIỀU

### Mảng 1 chiều

Kích thước Stack khi  
quá thiếu, lúc quá thừa



ThS. Trần Văn Thọ

123

## Hiện thực Stack dùng mảng

- ❖ Có thể tạo một Stack bằng cách khai báo một **mảng 1 chiều** với kích thước tối đa là MAXSIZE (ví dụ:  $\text{MAXSIZE} = 1000$ )
  - Stack có thể chứa tối đa MAXSIZE phần tử đánh số từ 0 đến MAXSIZE - 1
- ❖ Phần tử nằm ở đỉnh Stack sẽ có chỉ số là **Top** (*lúc đó trong Stack đang chứa Top + 1 phần tử*)
- ❖ Như vậy, để khai báo một Stack, ta cần một **mảng 1 chiều** Stack, và 1 biến số nguyên **Top** cho biết chỉ số của đỉnh Stack:

ThS. Trần Văn Thọ

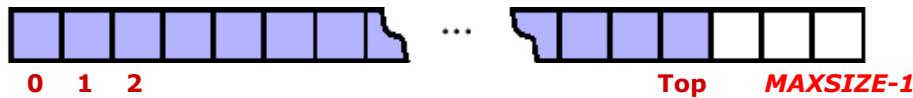
124

## KHAI BÁO STACK

Tạo cấu trúc dữ liệu cho Stack

`#define MAXSIZE 50 //Stack chứa 50 phần tử`

```
typedef int ItemType;
struct Stack
{
    int Top; //Số phần tử
    ItemType Info[MAXSIZE];
};
```



## THAO TÁC VỚI STACK

❖ Các thao tác trên Stack

initStack

isEmpty

isFull

push

pop

getTop

getSize

## Hiện thực Stack dùng mảng

❖ Ngăn xếp được cài đặt bằng mảng phải có:

- Một biến Top dùng để lưu vị trí con trỏ hiện tại.
- Một mảng dùng để lưu giá trị của ngăn xếp.

❖ Khởi tạo ngăn xếp rỗng:

- Ngăn xếp rỗng là ngăn xếp không chứa bất kỳ phần tử nào → Đỉnh ngăn xếp không trỏ tới phần tử nào.

❖ Kiểm tra ngăn xếp rỗng:

- Ngăn xếp rỗng khi Top = -1

❖ Kiểm tra ngăn xếp đầy:

- Ngăn xếp đầy khi Top = MAXSIZE

ThS. Trần Văn Thọ

127

## Hiện thực Stack dùng mảng

❖ Thêm một phần tử vào ngăn xếp:

- Kiểm tra ngăn xếp đầy.
- Top tăng lên 1 đơn vị.
- Phần tử tại vị trí Top bây giờ sẽ là phần tử cần thêm vào.

❖ Lấy một phần tử trong ngăn xếp:

- Kiểm tra ngăn xếp rỗng.
- Trả về phần tử tại vị trí Top.

❖ Xóa 1 phần tử khỏi ngăn xếp:

- Kiểm tra ngăn xếp rỗng.
- Top giảm đi 1 đơn vị.

ThS. Trần Văn Thọ

128

## Hiện thực Stack dùng mảng

### ☞ Nhận xét:

- Các thao tác trên đều làm việc với chi phí **O(1)**
- Việc cài đặt Stack thông qua mảng một chiều đơn giản và khá hiệu quả.
- Tuy nhiên, hạn chế lớn nhất của phương án cài đặt này là giới hạn về kích thước của Stack (**MAXSIZE**).
  - Giá trị của MAXSIZE có thể quá nhỏ so với nhu cầu thực tế hoặc quá lớn sẽ làm lãng phí bộ nhớ.
  - Nên hạn chế việc cài đặt ngăn xếp bằng mảng.

ThS. Trần Văn Thọ

129

## KHỞI TẠO STACK

```
void initStack(Stack &s)
{ //số phần tử ban đầu = 0
  s.Top = -1;
}
```

ThS. Trần Văn Thọ

130

## KIỂM TRA STACK RỖNG/ ĐẦY

```
int isEmpty(Stack s)
{
    return (s.Top == -1);
}

int isFull(Stack s)
{
    return (s.Top == MAXSIZE);
}
```

ThS. Trần Văn Thọ

131

## THÊM PHẦN TỬ VÀO STACK

```
int push(stack &s, ItemType x)
{
    if(isFull(s) == 1)
        return 0;
    s.Top++;
    s.Info[s.Top] = x;
    return 1;
}
```

Có thể viết gọn lại:

$s.Info[+s.Top] = x;$

ThS. Trần Văn Thọ

132

## LẤY PHẦN TỬ RA KHỎI STACK

```
int pop(Stack &s, ItemType &x)
{
    if (isEmpty(s) == 1)
        return 0;
    x = s.Info[s.Top];
    s.Top--;
    return 1;
}
```

Có thể viết gọn lại:

x = s.Info[s.Top--];