

NGĂN XẾP (STACK)

(1) **Lý thuyết:**

- Xem bài giảng CHƯƠNG 5 và giáo trình trang 90 - 103.

(2) **Bài tập tại lớp:**

1. *Viết CT đổi số từ hệ thập phân (cơ số 10) sang số ở hệ cơ số bất kỳ từ 2, 3, 4, ... 10:*

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#define MAXSIZE 50

struct STACK
{
    int top;//Đỉnh ngăn xếp
    int nodes[MAXSIZE];//Mỗi phần tử của mảng là một nút của ngăn xếp
};

void Initialize(STACK &stack) {
    stack.top = -1;
}

int Empty(STACK stack)
{
    return stack.top == -1 ? 1 : 0;
}

int Full(STACK stack)
{
    return stack.top == MAXSIZE - 1 ? 1 : 0;
}

void Push(STACK &stack, int x)
{
    if (Full(stack))
        printf("Ngan xep day!");
    else
        stack.nodes[++stack.top] = x;
}

int Pop(STACK &stack)
{
    if (Empty(stack))
        printf("Ngan xep rong!");
    else
        return stack.nodes[stack.top--];
}
```

```
int StackTop(STACK stack)
{
    if (Empty(stack))
        printf("Ngan xep rong!");
    else
        return stack.nodes[stack.top];
}

void main()
{
    STACK stack;
    int n, m, coso, du;
    Initialize(stack);
    printf("Nhap so: ");
    scanf("%d", &n);
    printf("Nhap co so: ");
    scanf("%d", &coso);
    m = n;
    while (m != 0)
    {
        du = m % coso;
        Push(stack, du);
        m /= coso;
    }
    printf("%d(%d) = ", n, 10);
    while (!Empty(stack))
        printf("%d", Pop(stack));
    printf("(%d)", coso);
    _getch();
}
```

Kết quả đổi chính xác với các hệ cơ số từ 2 => 10:

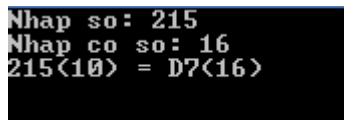
```
Nhap so: 215
Nhap co so: 2
215<10> = 11010111<2>
```

Nhưng không chính xác khi đổi sang hệ cơ số 11 => 16:

```
Nhap so: 215
Nhap co so: 16
215<10> = 137<16>
```

Cải tiến để có thể đổi sang hệ cơ số bất kỳ từ 2, 3, 4, ... 16 như sau:

```
void main()
{
    STACK stack;
    int n, m, coso, du;
    char s[16] = { '0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
    Initialize(stack);
    printf("Nhap so: ");
    scanf("%d", &n);
    printf("Nhap co so: ");
    scanf("%d", &coso);
    m = n;
    while (m != 0)
    {
        du = m%coso;
        Push(stack, s[du]);
        m /= coso;
    }
    printf("%d(%d) = ", n, 10);
    while (!Empty(stack))
        printf("%c", Pop(stack));
    printf("(%d)", coso);
    _getch();
}
```



```
Nhap so: 215
Nhap co so: 16
215(10) = D7(16)
```

2. **Viết CT tính trị của 1 biểu thức dạng hậu tố (postfix), xét biểu thức chỉ gồm 4 toán tử + - * / %, các toán hạng là các con số từ 0 => 9, không có khoảng trắng giữa các ký tự:**

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#define MAXSIZE 50

struct STACK
{
    int top;//Đỉnh ngăn xếp
    float nodes[MAXSIZE];//Mỗi phần tử của mảng là một nút của ngăn xếp
};

void Initialize(STACK &stack) {
    stack.top = -1;
}
```

```
int Empty(STACK stack)
{
    return stack.top == -1 ? 1 : 0;
}

int Full(STACK stack)
{
    return stack.top == MAXSIZE - 1 ? 1 : 0;
}

void Push(STACK &stack, float x)
{
    if (Full(stack))
        printf("Ngan xep day!");
    else
        stack.nodes[++stack.top] = x;
}

float Pop(STACK &stack)
{
    if (Empty(stack))
        printf("Ngan xep rong!");
    else
        return stack.nodes[stack.top--];
}

float StackTop(STACK stack)
{
    if (Empty(stack))
        printf("Ngan xep rong!");
    else
        return stack.nodes[stack.top];
}

int IsOperand(char c)
{
    return c >= '0' && c <= '9' ? 1 : 0;
}

int IsOperator(char c)
{
    return c == '+' || c == '-' || c == '*' || c == '/' || c == '%' ? 1 : 0;
}
```

```

float CalcPostfix(char postfix[])
{
    STACK stack;
    Initialize(stack);
    float operand1, operand2, res;
    for (int i = 0; i < strlen(postfix); i++)
    {
        if (IsOperand(postfix[i]))
            Push(stack, (float)(postfix[i] - '0'));
        else
        {
            operand2 = Pop(stack);
            operand1 = Pop(stack);
            switch (postfix[i])
            {
                case '+': res = operand1 + operand2; break;
                case '-': res = operand1 - operand2; break;
                case '*': res = operand1 * operand2; break;
                case '/': res = operand1 / operand2; break;
                case '%': res = operand1 + operand2; break;
            }
            Push(stack, res);
        }
    }
    return Pop(stack);
}

void main()
{
    char c, postfix[50];
    do{
        _flushall();
        printf("Nhap bieu thuc hau to: ");
        gets(postfix);
        printf("%s = %.2f\n", postfix, CalcPostfix(postfix));
        printf("Ban muon tiep tục? (c/k): ");
        c = getchar();
    } while (c == 'C' || c == 'c');
    _getch();
}

```

```

Nhap bieu thuc hau to: 63*1-
63*1- = 17.00
Ban muon tiep tục? <c/k>: c
Nhap bieu thuc hau to: 123*+
123*+ = 7.00
Ban muon tiep tục? <c/k>: k

```

(3) Bài tập về nhà:

- Viết CT đảo ngược 1 chuỗi ký tự.