HÀNG ĐỢI (QUEUE)

(1) Lý thuyết:

- Xem bài giảng CHƯƠNG 6 và giáo trình trang 104 - 121.

(2) Sửa bài tập về nhà buổi 13:

Viết CT đảo ngược 1 chuỗi ký tự sử dụng ngăn xếp kiểu liên kết.

```
Nhap chuoi: Cau Truc Du Lieu
Chuoi dao: ueiL uD curT uaC
```

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<string.h>
#define MAXSIZE 256

//Khai báo cấu trúc 1 nút của ngăn xếp
struct node
{
	char info; //chứa nội dung của nút
	node *next; //con trỏ chỉ nút kế tiếp trong ngăn xếp
};

//Khai báo kiểu con trỏ chỉ đến nút
typedef node *NODEPTR;

//Cấp phát biến động làm 1 nút cho ngăn xếp
NODEPTR GetNode()
{
	NODEPTR p;
	p = (NODEPTR)malloc(sizeof(node));
	return p;
}

//Giải phóng biến động đã cấp phát trước đó
void FreeNode(NODEPTR p)
{
	free(p);
}

//Khởi động ngăn xếp
void Initialize(NODEPTR &pstack)
{
	pstack = NULL;
}

//Kiểm tra ngăn xếp có bị rỗng hay không
int Empty(NODEPTR pstack)
{
	return (pstack == NULL ? 1 : 0);
}
```

```
//Thêm nút mới (có nội dung c) vào đầu ngăn xếp
void Push(NODEPTR &pstack, char c)
{
	NODEPTR p = GetNode();
	p->info = c;
	p->next = pstack;
	pstack = p;
}

//Xoá nút tại đỉnh ngăn xếp
char Pop(NODEPTR &pstack)
{
	if (Empty(pstack))
		printf("Ngan xep rong\n");
	else
	{
		NODEPTR p = pstack;
		char c = p->info;
		pstack = p->next;
		p->next = NULL;
		FreeNode(p);
		return c;
	}
}

char StackTop(NODEPTR pstack)
{
	if (Empty(pstack))
		printf("Ngan xep rong\n");
	else
		return pstack->info;
}

//Chương trình đảo ngược 1 chuỗi ký tự
void main()
{
	NODEPTR pstack;
	Initialize(pstack);
	char s[MAXSIZE];
	printf("Nhap chuoi: ");
	gets(s);
	for (int i = 0; i < strlen(s); i++)
		Push(pstack, s[i]);
	printf("Chuoi dao: ");
	while (!Empty(pstack))
		printf("%c",Pop(pstack));
	_getch();
}
```

(3) Bài tập tại lớp:

1) Viết chương trình cài đặt hàng đợi theo kiểu kế tiếp:

- ***Phương pháp di chuyển tịnh tiến:***

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

#define MAXSIZE 5
struct QUEUE
{
    int rear;//Cuối hàng đợi
    int nodes[MAXSIZE];//Mỗi nút hàng đợi là một phần tử trên mảng một chiều
};

//Khởi động hàng đợi
void Initialize(QUEUE &queue)
{
    queue.rear = -1;
}

//Kiểm tra hàng đợi có bị rỗng không?
int Empty(QUEUE queue) {
    return queue.rear == -1 ? 1 : 0;
}

//Kiểm tra hàng đợi có bị đầy không?
int Full(QUEUE queue)
{
    return queue.rear == MAXSIZE - 1 ? 1 : 0;
}

//Thêm nút có nội dung là x vào cuối hàng đợi
void Insert(QUEUE &queue, int x)
{
    if (Full(queue))
        printf("Hang doi bi day\n");
    else
        queue.nodes[++queue.rear] = x;
}

//Xoá nút ở đầu hàng đợi
int Remove(QUEUE &queue) {
    if (Empty(queue))
        printf("Hang doi bi rong\n");
    else
    {
        int x = queue.nodes[0];
        //Dời các nút còn lại của hàng đợi xuống dưới một vị trí
        for (int i = 0; i < queue.rear; i++)
            queue.nodes[i] = queue.nodes[i + 1];
        queue.rear--;
        return x;
    }
}
```

```
//Duyệt hàng đợi từ nút đầu đến nút cuối
void Traverse(QUEUE queue)
{
    if (Empty(queue))
            printf("Hang doi rong\n");
    else
    {
            for (int i = 0; i <= queue.rear;i++)
                    printf("%d\t",queue.nodes[i]);
            printf("\n");
    }
}

void main()
{
    QUEUE queue;
    Initialize(queue);
    int i = 1;
    while (!Full(queue))
    {
            Insert(queue, i);
            printf("Da them nut %d o cuoi hang doi\n", i);
            if (Full(queue))
            {
                    printf("Hang doi bi day\n");
                    break;
            }
            i++;
    }
    printf("Duyet hang doi:\n");
    Traverse(queue);
    while (!Empty(queue))
            printf("Da xoa nut %d o dau hang doi\n", Remove(queue));
    printf("Hang doi rong");
    _getch();
}
```

- ***Phương pháp di chuyển vòng:***

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

#define MAXSIZE 5
struct QUEUE
{
    int front;//Trước nút đầu hàng đợi
    int rear;//Ngay nút cuối hàng đợi
    int nodes[MAXSIZE];//Mỗi nút hàng đợi là một phần tử trên mảng một chiều
};

//Khởi động hàng đợi
void Initialize(QUEUE &queue)
{
    queue.front = queue.rear = MAXSIZE - 1;
}

//Kiểm tra hàng đợi có bị rỗng không?
int Empty(QUEUE queue)
{
    return (queue.front == queue.rear ? 1 : 0);
}

//Kiểm tra hàng đợi có bị đầy không?
int Full(QUEUE queue)
{
 return queue.front == queue.rear + 1 || queue.front == 0 && queue.rear == MAXSIZE -1 ? 1: 0;
}

//Thêm nút có nội dung là x vào cuối hàng đợi
void Insert(QUEUE &queue, int x)
{
    if (Full(queue))
            printf("Hang doi bi day\n");
    else
    {
            if (queue.rear == MAXSIZE-1)
                    queue.rear = 0;
            else
                    queue.rear++;
            queue.nodes[queue.rear] = x;
    }
}
```

```
//Xoá nút ở đầu hàng đợi
int Remove(QUEUE &queue)
{
    if (Empty(queue))
            printf("Hang doi rong\n");
    else
    {
            if (queue.front == MAXSIZE-1)
                    queue.front = 0;
            else
                    queue.front++;
            return  queue.nodes[queue.front];
    }
}
//Duyệt hàng đợi từ nút đầu đến nút cuối
void Traverse(QUEUE queue)
{
    if (Empty(queue))
            printf("Hang doi rong\n");
    else
    {
            int i;
            if (queue.front = MAXSIZE - 1)
                    i = 0;
            else
                    i = queue.front + 1;
            //In giá trị các nút từ đầu đến kế cuối
            while (i!=queue.rear)
            {
                    printf("%d\t",queue.nodes[i]);
                    if (i == MAXSIZE - 1)
                            i = 0;
                    else
                            i++;
            }
            //In giá trị nút cuối
            printf("%d\n",queue.nodes[i]);
    }
}
```

(4) Bài tập về nhà:

- ***Viết chương trình cài đặt hàng đợi theo kiểu liên kết.***

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<malloc.h>

//Khai báo cấu trúc 1 nút trong hàng đợi
struct node
{
    int info; //chứa nội dung của nút
    node *next; //con trỏ chỉ nút kế tiếp trong hàng đợi
};

//Khai báo kiểu con trỏ chỉ đến nút trong hàng đợi
typedef node *NODEPTR;

//Cấp phát biến động làm 1 nút cho hàng đợi
NODEPTR GetNode()
{
    NODEPTR p;
    p = (NODEPTR)malloc(sizeof(node));
    return p;
}

//Giải phóng biến động đã cấp phát trước đó
void FreeNode(NODEPTR p)
{
    free(p);
}

//Khởi động hàng đợi
void Initialize(NODEPTR &pqfront,NODEPTR &pqrear)
{
    pqfront = pqrear = NULL;
}

//Kiểm tra hàng đợi có bị rỗng hay không
int Empty(NODEPTR pqfront, NODEPTR pqrear)
{
    return pqfront == NULL&&pqrear == NULL ? 1 : 0;
}

//Thêm 1 nút vào cuối hàng đợi
void Insert(NODEPTR &pqfront, NODEPTR &pqrear, int x)
{
    NODEPTR p = GetNode();
    p->info = x;
    p->next = NULL;
    if (Empty(pqfront, pqrear))
            pqfront = pqrear = p;
    else
    {
            pqrear->next = p;
            pqrear = p;
    }
}
```

```
//Xóa nút đầu hàng đợi
int Remove(NODEPTR &pqfront, NODEPTR &pqrear)
{
	if (Empty(pqfront, pqrear))
		printf("Hang doi rong\n");
	else
	{
		NODEPTR p = pqfront;
		int x = p->info;
		if (p->next == NULL)
			pqfront = pqrear = NULL;
		else
			pqfront = p->next;
		FreeNode(p);
		return x;
	}
}

//Duyệt các nút trong hàng đợi
void Traverse(NODEPTR pqfront, NODEPTR pqrear)
{
	if (Empty(pqfront, pqrear))
		printf("Hang doi rong\n");
	else
	{
		NODEPTR p = pqfront;
		while (p != pqrear)
		{
			printf("%d\t",p->info);
			p = p->next;
		}
		printf("%d\t", p->info);//In nút cuối
	}
}

//Xóa tất cả các nút trong hàng đợi
void ClearQueue(NODEPTR &pqfront, NODEPTR &pqrear)
{
	if (Empty(pqfront, pqrear))
		printf("Hang doi rong\n");
	else
	{
		int x;
		while (!Empty(pqfront, pqrear))
		{
			x = Remove(pqfront, pqrear);
			printf("Da xoa nut %d dau hang doi.\n",x);
		}
	}
}
```

```
void main()
{
    NODEPTR pqfront,pqrear;
    Initialize(pqfront,pqrear);
    int chon,x;
    do
    {
            printf("\n-----CHUONG TRINH XU LY HANG DOI-----\n");
            printf("1. Them 1 nut vao cuoi hang doi\n");
            printf("2. Xoa nut dau hang doi\n");
            printf("3. Xem gia tri nut dau hang doi\n");
            printf("4. Xem gia tri nut cuoi hang doi\n");
            printf("5. Duyet cac nut trong hang doi\n");
            printf("6. Xoa tat ca cac nut trong hang doi\n");
            printf("0. Thoat CT\n");
            printf("-------------------------------------\n");
            printf("Ban chon: ");
            scanf("%d",&chon);
            switch (chon)
            {
            case 0:
                    break;
            case 1:
                    printf("Nhap gia tri nut can them: ");
                    scanf("%d",&x);
                    Insert(pqfront, pqrear, x);
                    printf("Hang doi sau khi them nut:\n ");
                    Traverse(pqfront, pqrear);
                    break;
            case 2:
                    if (Empty(pqfront, pqrear))
                            printf("Hang doi rong\n");
                    else
                    {
                            Remove(pqfront, pqrear);
                            printf("Hang doi sau khi xoa nut:\n ");
                            Traverse(pqfront, pqrear);
                    }
                    break;
            case 3:
                    if (Empty(pqfront, pqrear))
                            printf("Hang doi rong\n");
                    else
                            printf("Gia tri nut dau hang doi la: %d\n", pqfront->info);
                    break;
            case 4:
                    if (Empty(pqfront, pqrear))
                            printf("Hang doi rong\n");
                    else
                            printf("Gia tri nut cuoi hang doi la: %d\n", pqrear->info);
                    break;
```

```
            case 5:

                if (Empty(pqfront, pqrear))
                    printf("Hang doi rong\n");
                else
                    Traverse(pqfront,pqrear);
                break;
            case 6:
                if (Empty(pqfront, pqrear))
                    printf("Hang doi rong\n");
                else
                {
                    ClearQueue(pqfront, pqrear);
                    printf("Da xoa tat ca cac nut trong hang doi.\n");
                }
                break;
            default:
                printf("Ban chon sai. Moi chon lai.\n");
            }
    } while (chon!=0);
    _getch();
}
```