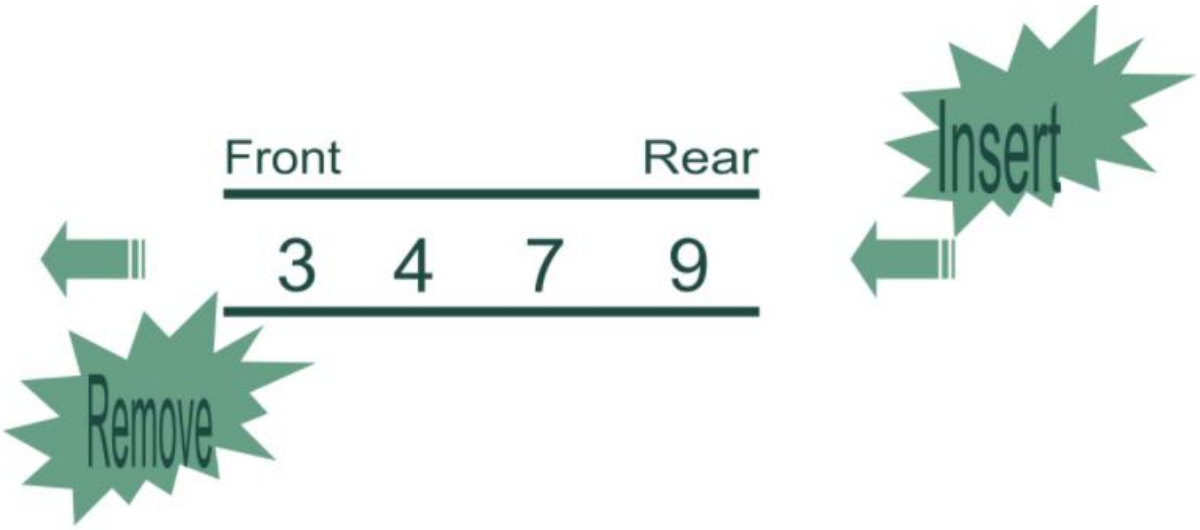
Cấu trúc dữ liệu hàng đợi queue trong c/c++



Hai thao tác cơ bản với cấu trúc dữ liệu hàng đợi queue

- **Add(q,x):** thêm phần tử x vào hang đợi queue
- **Remove(q):** lấy phần tử (hay đối tượng) được thêm ở đầu queue ra khỏi hàng đợi và trả về giá trị của nó, đồng thời thực hiện xóa phần tử đó đi.

Cấu trúc hàng đợi queue với mảng

```
#define MAX 100 // so phan tu toi da 100
typedef struct queue
{
        int A[MAX]; // khai bao mot hang doi bang mang co so luong phan tu MAX=100
        int front, rear; // khai bao chi so dau và chi so cuoi
};
typedef struct queue QUEUE;
```

Cấu trúc dữ liệu hàng đợi queue với danh sách liên kết

Khai báo các node và cấu trúc queue với dslk

```
struct node{
	int data; // thanh phan du lieu kiêu so nguyen
	node *next; // con trỏ next
};
typedef struct node NODE;
```

```
struct queue{  // khai bao mot hang doi queue 2 thanh phan NODE * front
	NODE *front;
	NODE *rear;
};
typedef struct queue QUEUE;
```

Hàm khởi tạo queue

```
void Init(QUEUE &q){
	//gan front va rear ve NULL
	q.front = NULL;
	q.rear = NULL;
}
```

Hàm tạo một node trong queue

```
NODE* CreateNode (int x) {
	NODE *p;
	p = new NODE;
	//neu p = NULL thi ko du bo nho cap phat
	if (p==NULL) {
	cout <<"KHONG DU BO NHO!";
	return NULL;
	}
	//gan data bang X
	p->data=x;
```

```
        //gan con tro next bang NULL
        p->next=NULL;
        //tra ve node p
        return p;
}
```

Hàm kiểm tra rỗng trong queue

```
int IsEmpty(QUEUE q){
        //neu front bang NULL thi queue rong
        if (q.front == NULL){
        return 1;
}
        //nguoc lai tra ve 0
        return 0;
}
```

Thao tác add trong queue

```
void Add (QUEUE &q, NODE *NewNode){
        //neu hang doi rong thi them node moi vao ca dau va cuoi hang doi
        if(q.front == NULL){
        q.front = NewNode;
        q.rear = NewNode;
        }
        else{//nguoc lai them cuoi hang doi
        //dat con tro next cua phan tu cuoi ve NewNode
                q.rear->next = NewNode;
                //gan lai phan tu cuoi cua danh sach
                q.rear = NewNode;
}
}
```

Thao tác remove trong queue

```
int Remove(QUEUE &q){
	int x;
	NODE *p = NULL;
	//neu queue khong rong thuc hien lay phan tu dau queue
	if (!IsEmpty(q)){
		//gan node p bang phan tu dau tien cua queue
		p = q.front;
		//gan du lieu cua node p vao x
		x = p->data;
		//xoa di node dau tien cua queue
		q.front = q.front->next;
		delete p;
		//neu front bang NULL thi gan luon rear bang NULL
		if (q.front==NULL){
			q.rear = NULL;
		}
	}
	//tra ve du lieu x vua lay ra
	return x;
}
```

Nhập xuất node vào trong queue

+ Hàm nhập N node vào trong queue

```
void Input(QUEUE &q, int n){
	//duyet N lan
	for(int i = 0; i< n; i++){
	//nhap phan tu vao bien x
	int x;
	cuot << Nhap phan tu thu " << i << endl;
	cin >> x;
	cout << "\n";
	//tao node p co du lieu la x
```

```
        NODE *p;
        p = CreateNode(x);
        //them node p vao queue
        Add(q,p);
        }
}
```

Hàm xuất các node trong queue ra màn hình (nhưng không xóa)

```
void Output(QUEUE q){
        //duyet tu dau den cuoi hang doi
        for(NODE *p = q.front; p!= NULL; p=p->next){
        //hien thi data cua cac node
        Cout <<"\n" << p->data;
        }
}
```

Chương trình:

```
#include <iostream>
Using namespace std;

struct node{
        int data;
        node *next;
};
typedef struct node NODE;
struct queue{
        NODE *front;
        NODE *rear;
};
typedef struct queue QUEUE;
void Init(QUEUE &q){
        //gan front va rear ve NULL
```

```
	q.front = NULL;
	q.rear = NULL;
}
NODE* CreateNode (int x) {
	NODE *p;
	p = new NODE;
	//neu p = NULL thi ko du bo nho cap phat
	if (p==NULL) {
	cout << “KHONG DU BO NHO!";
	return NULL;
	}
	//gan data bang X
	p->data=x;
	//gan con tro next bang NULL
	p->next=NULL;
	//tra ve node p
	return p;
}
int IsEmpty(QUEUE q){
	//neu front bang NULL thi queue rong
	if (q.front == NULL){
	return 1;
	}
	//nguoc lai tra ve 0
	return 0;
}
void Add (QUEUE &q, NODE *NewNode){
	//neu hang doi rong thi them node moi vao ca dau va cuoi hang doi
	if(q.front == NULL){
	q.front = NewNode;
	q.rear = NewNode;
	}
	else{//nguoc lai them cuoi hang doi
```

```
		//dat con tro next cua phan tu cuoi ve NewNode
		q.rear->next = NewNode;
		//gan lai phan tu cuoi cua danh sach
		q.rear = NewNode;
		}
}
int Remove(QUEUE &q){
		int x;
		NODE *p = NULL;
		//neu queue khong rong thuc hien lay phan tu dau queue
		if (!IsEmpty(q)){
			//gan node p bang phan tu dau tien cua queue
			p = q.front;
			//gan du lieu cua node p vao x
			x = p->data;
			//xoa di node dau tien cua queue
			q.front = q.front->next;
			delete p;
			//neu front bang NULL thi gan luon rear bang NULL
			if (q.front==NULL){
				q.rear = NULL;
			}
		}
		//tra ve du lieu x vua lay ra
		return x;
}
void Input(QUEUE &q, int n){
		//duyet N lan
		for(int i = 0; i< n; i++){
		//nhap phan tu vao bien x
		int x;
		cout << "Nhap phan tu thu  "<< i;
		cin >> x;
```

```
        //tao node p co du lieu la x
        NODE *p;
        p = CreateNode(x);
        //them node p vao queue
        Add(q,p);
        }
}
void Output(QUEUE q){
        //duyet tu dau den cuoi hang doi
        for(NODE *p = q.front; p!= NULL; p=p->next){
        //hien thi data cua cac node
        Cout << " \t" <<p->data);
        }
}
int main(){
        //tao moi queue
        QUEUE q;
        //khoi tao queue
        Init(q);
        //nhap n phan tu vao queue
        int n;
        cout <<"NHAP N: ";
        cin >> n;
        Input(q,n);
        //hien thi phan tu trong queue
        cout << "CAC PHAN TU TRONG HANG DOI LA\n";
        Output(q);
        //them mot node co du lieu bang 66 vao hang doi
        int x = 66;
        NODE *p = CreateNode(x);
        Add(q,p);
        cout <<"\nHANG DOI SAU KHI THEM NODE \n" <<x;
        Output(q);
```

```
        //thuc hien lay phan tu trong hang doi ra
        int k = Remove(q);
        cout << "\nHANG DOI SAU KHI REMOVE %d \n" <<k;
        Output(q);
}
```

Bài tập áp dụng: Xử lý yêu cầu đề bài bằng queue cài đặt bằng danh sách liên kết

Trên ngôn ngữ C/C++, viết chương trình thực hiện yêu cầu sau: Khai báo cấu trúc date gồm các thông tin: ngày, tháng, năm. Khai báo cấu trúc máy tính cần bốc xếp trên băng chuyền bao gồm các thông tin: Mã máy, tên máy, ngày xuất hàng, giá xuất hàng đi (đơn vị triệu đồng):

1.Viết các hàm nhập vào từ bàn phím giá trị của một biến cấu trúc máy tính, hiển thị lên màn hình

2.Nhập vào 1 số nguyên dương n, sử dụng cấu trúc dữ liệu hàng đợi (queue) nhập vào băng chuyền gồm n máy tính cần bốc xếp.

3.Hiển thị lên màn hình thông tin tất cả các máy tính trong hàng đợi có mã hàng hóa (trường ID) là “007”.

4.Hiển thị lên màn hình thông tin máy tính đã bốc xếp (hiện tại băng chuyền đã rỗng).