Các lệnh khai báo trong c

++++ queue:

#include <stdlib.h>  
#include <stdio.h>  
  
// Định nghĩa cấu trúc Node  
typedef struct node {  
int data;  
struct node\* next;  
} Node;  
  
// Định nghĩa queue  
typedef struct queue {  
Node\* front;  
Node\* rear;  
} Queue;  
  
// Khởi tạo queue  
void initQueue(Queue\* q) {  
q->front = q->rear = NULL;  
}  
  
// Kiểm tra queue rỗng  
int isEmpty(Queue\* q) {  
return q->front == NULL;  
}  
  
// Thêm phần tử vào queue  
void enqueue(Queue\* q, int data) {  
Node\* newNode = (Node\*)malloc(sizeof(Node));  
newNode->data = data;  
newNode->next = NULL;  
  
if (isEmpty(q)) {  
q->front = q->rear = newNode;  
} else {  
q->rear->next = newNode;  
q->rear = newNode;  
}  
}  
  
// Lấy phần tử đầu tiên của queue  
int dequeue(Queue\* q) {  
if (isEmpty(q)) {  
printf("Queue is empty\n");  
return -1;  
} else {  
int data = q->front->data;  
Node\* temp = q->front;  
  
if (q->front == q->rear) {  
q->front = q->rear = NULL;  
} else {  
q->front = q->front->next;  
}  
  
free(temp);  
return data;  
}  
}  
  
// Lấy số lượng phần tử trong queue  
int size(Queue\* q) {  
int count = 0;  
Node\* temp = q->front;  
  
while (temp != NULL) {  
++count;  
temp = temp->next;  
}  
  
return count;  
}  
  
int main() {  
Queue q;  
  
initQueue(&q);  
enqueue(&q, 10);  
enqueue(&q, 20);  
enqueue(&q, 30);  
  
printf("Queue size is %d\n", size(&q));  
printf("%d dequeued from queue\n", dequeue(&q));  
printf("%d dequeued from queue\n", dequeue(&q));  
printf("Queue size is %d\n", size(&q));  
  
return 0;  
}

+++stack

1. Để khởi tạo một stack:  
  
```  
#include <stdlib.h>  
#include <stdio.h>  
  
int main() {  
int size = 10;  
int\* stack = (int\*)malloc(size \* sizeof(int));  
int top = -1;  
return 0;  
}  
```  
  
2. Để đặt một phần tử lên đỉnh của stack:  
  
```  
void push(int \*stack, int data, int \*top, int size) {  
if(\*top == size - 1) {  
printf("Stack is full.\n");  
return;  
}  
\*top = \*top + 1;  
stack[\*top] = data;  
}  
```  
  
3. Để lấy phần tử trên đỉnh của stack:  
  
```  
int pop(int \*stack, int \*top) {  
if(\*top == -1) {  
printf("Stack is empty.\n");  
return NULL;  
}  
int data = stack[\*top];  
\*top = \*top - 1;  
return data;  
}  
```  
  
4. Để kiểm tra xem stack có rỗng hay không:  
  
```  
int isEmpty(int top) {  
if(top == -1) {  
return 1;  
}  
else {  
return 0;  
}  
}