Aim

Implement queue using two stacks.

Algorithm

- Enqueue: push onto stack1.
- Dequeue: if stack2 empty, pop all from stack1 into stack2, then pop from stack2.

C Code

```c
#include <stdio.h>
#include <stdlib.h>

#define SIZE 100

struct Stack {
    int arr[SIZE];
    int top;
};

void init(struct Stack* s) { s->top=-1; }
int isEmpty(struct Stack* s) { return s->top==-1; }
void push(struct Stack* s,int x){ s->arr[++s->top]=x; }
int pop(struct Stack* s){ return isEmpty(s)?-1:s->arr[s->top--]; }

struct Queue {
    struct Stack s1,s2;
};

void enqueue(struct Queue* q,int x){ push(&q->s1,x); }
int dequeue(struct Queue* q){
    if(isEmpty(&q->s2))
        while(!isEmpty(&q->s1)) push(&q->s2,pop(&q->s1));
    return pop(&q->s2);
}

int main(){
    struct Queue q; init(&q.s1); init(&q.s2);
    enqueue(&q,1); enqueue(&q,2); enqueue(&q,3);
    printf("Dequeued: %d\n",dequeue(&q));
    printf("Dequeued: %d\n",dequeue(&q));
    return 0;
}
```

Input

Enqueue 1,2,3 → Dequeue twice.

Output

```
Dequeued: 1
Dequeued: 2
```

Result

Queue implemented using two stacks.