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

struct sNode {
   int data;
   struct sNode* next;
};

void push(struct sNode** top_ref, int new_data);

int pop(struct sNode** top_ref);

struct queue {
   struct sNode* stack1;
   struct sNode* stack2;
};

void enQueue(struct queue* q, int x)
{
   push(&q->stack1, x);
}

int deQueue(struct queue* q)
{
   int x;

   if (q->stack1 == NULL && q->stack2 == NULL) {
      printf("Q is empty");
      getchar();
      exit(0);
   }
```

```
    if (q->stack2 == NULL) {
        while (q->stack1 != NULL) {
            x = pop(&q->stack1);
            push(&q->stack2, x);
        }
    }

    x = pop(&q->stack2);
    return x;
}

void push(struct sNode** top_ref, int new_data)
{
    struct sNode* new_node = (struct sNode*)malloc(sizeof(struct sNode));
    if (new_node == NULL) {
        printf("Stack overflow \n");
        getchar();
        exit(0);
    }

    new_node->data = new_data;

    new_node->next = (*top_ref);

    (*top_ref) = new_node;
}

int pop(struct sNode** top_ref)
{
    int res;
```

```
    struct sNode* top;

    if (*top_ref == NULL) {
        printf("Stack underflow \n");
        getchar();
        exit(0);
    }
    else {
        top = *top_ref;
        res = top->data;
        *top_ref = top->next;
        free(top);
        return res;
    }
}

int main()
{
    struct queue* q = (struct queue*)malloc(sizeof(struct queue));
    q->stack1 = NULL;
    q->stack2 = NULL;
    enQueue(q, 110);
    enQueue(q, 220);
    enQueue(q, 1233);
    enQueue(q, 1120);
    enQueue(q, 1610);
    printf("%d ", deQueue(q));
    printf("%d ", deQueue(q));
```

```
    printf("%d ", deQueue(q));
    printf("%d ", deQueue(q));
    printf("%d ", deQueue(q));

    return 0;
}
```