queue-using-two-stack.c

```
//write a program to implement queue using two stack.
#include <stdio.h>
#include <stdlib.h>

typedef struct Stack{
   int data;
   struct Stack *next;
}stack;

int len(stack *st){
   int count = 0;
   while(st){
      st = st->next;
      count++;
   }
   return count;
}

int isEmpty(stack *st){
   return (st == NULL);
}

void push(stack **st, int data){
   stack *new_node = (stack*)malloc(sizeof(stack));
   new_node->data = data;
   new_node->next = *st;
   *st = new_node;
}

void enqueue(stack **st, int data){
   push(st,data);
}

int pop(stack **st){
   stack *temp = *st;
   *st = (*st)->next;
   int data = temp->data;
   free(temp);
   return data;
}

int dequeue(stack **st){
   if(!*st){
      return -1;
   }
   stack *st1 = NULL;
   while(*st != NULL){
      push(&st1,pop(st));
   }
   int data = pop(&st1);
   while(st1 != NULL){
      push(st,pop(&st1));
   }
   return data;
}
```

```
void display(stack *st){
   while(st){
      printf("%d ",st->data);
      st = st->next;
   }
   printf("\n");
}

void main(){
   stack *q = NULL;
   if(isEmpty(q)){
      printf("The queue is empty.\n");
   }else{
      printf("The queue is not empty.\n");
   }
   printf("the length of the queue is %d\n",len(q));
   enqueue(&q,10);
   printf("the length of the queue is %d\n",len(q));
   enqueue(&q,102);
   enqueue(&q,15);
   enqueue(&q,13);
   enqueue(&q,12);
   display(q);
   printf("the length of the queue is %d\n",len(q));
   dequeue(&q);
   dequeue(&q);
   dequeue(&q);
   display(q);
}
```

OUTPUT

```
PS S:\WorkSpace\CollegeWork\DataStructure\Temp> gcc .\queue-using-two-stack.c
PS S:\WorkSpace\CollegeWork\DataStructure\Temp> ./a
The queue is empty.
the length of the queue is 0
the length of the queue is 1
12 13 15 102 10
the length of the queue is 5
12 13
PS S:\WorkSpace\CollegeWork\DataStructure\Temp>
```