Stacks and Queues- implementation

Code for manipulations of stacks and queues. Functions are provided for push, pop, enqueue and dequeue operations. The stack is indexed by stacktop, while the queue is being maintained by two indices – qfront and qback.

```
#include <stdio.h>

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


void push(struct node**stacktop,int d );
int pop(struct node**stacktop);
void enqueue(struct node**qfront,int d , struct node**qback);
int dequeue(struct node**qfront, struct node**qback);
```

```
void  main( ) {
  int num;
  int dd, ii;
  struct node* first = (struct node*) (malloc(sizeof(struct node)));
  struct node* front = (struct node*) (malloc(sizeof(struct node)));
  struct node* back = (struct node*) (malloc(sizeof(struct node)));
  front = NULL;
  back= NULL;
  first = NULL;
```

```
for (ii= 1; ii<=3; ii++) {
        printf("\n next number? =");
        scanf("%d",&dd);
        push(&first,dd);
}

for (ii= 1; ii<=4; ii++)
{
        dd = pop(&first);
        if (dd !=-999)
         printf("\n popped number =%d", dd);
        printf("\n\n");
}

dd=0;



for (ii= 1; ii<=4; ii++)
{
        printf("\n give number for queue? ="); scanf("%d",&dd);
        enqueue(&front,dd,&back);
        printf("\n front value is %d",front->data);
}
for (ii= 1; ii<=5; ii++)
{
        dd = dequeue(&front,&back);
        if (dd != -999)
        printf("\n  number from queue =%d",dd);
 }

printf("\n");
}//end of main function
```

```
//  stack functions:push

void push(struct node**stacktop,int d )
{
  struct node* pNew = (struct node*) (malloc(sizeof(struct node)));

       pNew-> data =  d ;
       pNew->next  =  *stacktop;
      *stacktop  =  pNew ;
}
```

```
//Pop Function returns the data from top element, and frees that
//element from the stack.
int pop(struct node* *stacktop)
{
    struct node* temp; int d;
    if(*stacktop== NULL)
      {
         printf("\nstack empty\n");
            d = -999;
       }
     else
      {
            d  =  temp->data;
             temp = *stacktop;
            *stacktop  =  temp->next ;
            free (temp);
        }
      return d;
}
```

```
// Enqueue function : qfront and qback point to first element of the
queue
//and the last element of queue respectively.
```