Experiment 5.5
Date 05-11-2022

REVERSE A QUEUE USING STACK

Aim

Write a program to reverse the contents of a queue using stack

Algorithm

```
Enqueue(item)
    Step 1: if(front == rear)
                print "Queue overflow"
    Step 2: else
        Step 2.1: if(front == -1) front =0
        Step 2.2: rear = rear + 1
        Step 2.3: queue[rear] = element
    Step 3: Stop

push(value)
    Step 1: if(TOP == size-1)
                return "All items of queue filled in stack"
    Step 2: else
        Step 2.1: TOP = TOP + 1
        Step 2.2: A[TOP] = item
    Step 3: Stop

Dequeue()
    Step 1: if (front == -1)
                print "Empty"
    Step 2: else
        Step 2.1: push(A[front]
        Step 2.2: front = front + 1
        Step 2.3: if(front > rear)
                    front = rear = -1
    Step 3: Stop

pop()
Step 1. if (TOP == -1)
            print "Reverse printed"
```

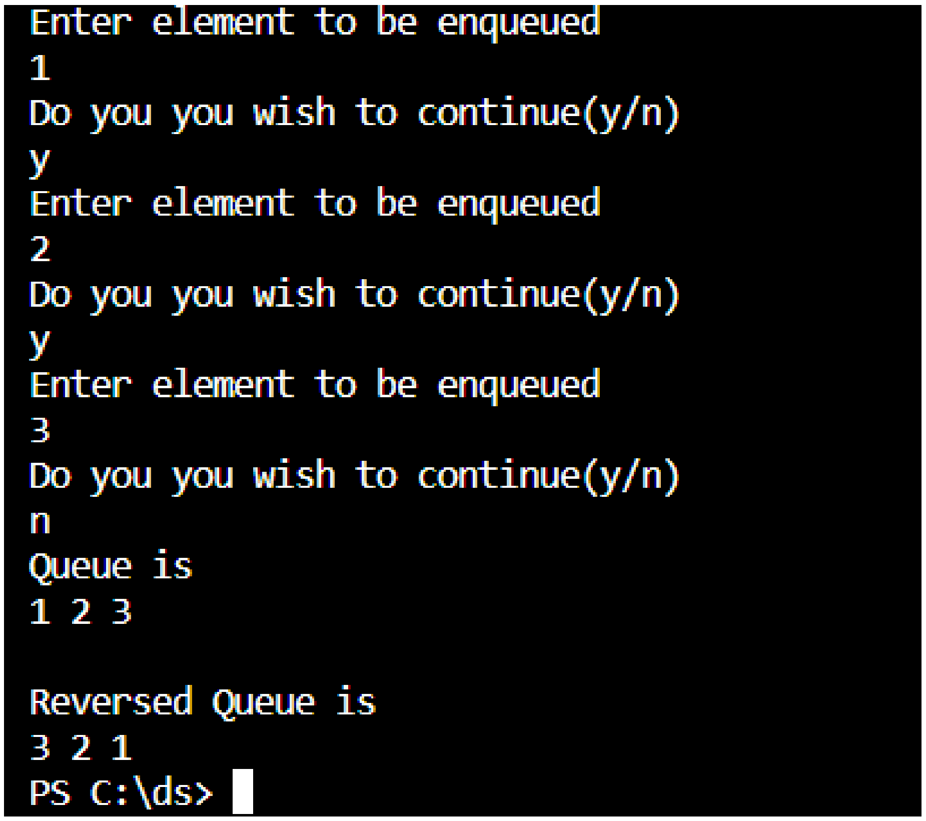
```
Step 2. else
    Step 2.1: temp = A[TOP]
    Step 2.2: TOP = TOP -1
    Step 2.3: return temp
```

Program Code

```c
#include<stdio.h>
#include<stdlib.h>
struct ptr
{
    int data;
    struct ptr*link;
};
struct ptr*head;
struct ptr*front;
struct ptr*rear;
void enqueue(int e)
{
    struct ptr*newptr=(struct ptr*)malloc(sizeof(struct ptr));
    if(front==NULL)
    {
        newptr->data=e;
        newptr->link=NULL;
        front=rear=newptr;
    }
    else
    {
        rear->link=newptr;
        rear=newptr;
        rear->data=e;
        rear->link=NULL;
    }
}
int dequeue()
{
    int item;
    struct ptr*temp;
    if(front==NULL)
        printf("Queue underflow\n");
    else
    {
        item=front->data;
        temp=front;
        front=front->link;
        free(temp);
    }
    return item;
}
void push(int e)
{
    struct ptr*newptr=(struct ptr*)malloc(sizeof(struct ptr));
    if(head==NULL)
    {
        newptr->data=e;
        newptr->link=NULL;
        head=newptr;
    }
    else
    {
        newptr->data=e;
        newptr->link=head;
        head=newptr;
    }
}
int pop()
{
    int item;
    struct ptr*temp;
    if(head==NULL)
        printf("Empty stack\n");
    else
    {
        item=head->data;
        temp=head;
        head=head->link;
        free(temp);
    }
    return item;
}
void display()
{
    int i;
    struct ptr*temp;
    temp=front;
    if(temp==NULL)
```

```
        printf("Empty queue\n");
    else
    {
        printf("Queue is \n");
        while(temp!=NULL)
        {
            printf("%d ",temp->data);
            temp=temp->link;
        }
        printf("\n");
    }
}
void main()
{
    int n=0,i,e,d;
    char ch='Y';
    while(ch=='y'||ch=='Y')
    {
        printf("Enter element to be enqueued\n");
        scanf("%d",&e);
        n++;
        enqueue(e);
        printf("Do you you wish to continue(y/n)\n");
        scanf(" %c",&ch);
    }
    display();
    for(i=0;i<n;i++)
    {
        d=dequeue();
        push(d);
    }
    for(i=0;i<n;i++)
    {
        d=pop();
        enqueue(d);
    }
    printf("\nReversed ");
    display();
}
```

Sample Input and Output



```
Enter element to be enqueued
1
Do you you wish to continue(y/n)
y
Enter element to be enqueued
2
Do you you wish to continue(y/n)
y
Enter element to be enqueued
3
Do you you wish to continue(y/n)
n
Queue is
1 2 3

Reversed Queue is
3 2 1
PS C:\ds>
```

Result

Program executed successfully