**[201533661 이승수’s Data Structure Lab1-2]**

**[Lab1-2]**

**Code)**

#include <stdio.h>

#include <stdlib.h>

/\*Global Variables & define\*/

#define size 10 //queue size

int queue[size]={0,};

int front=-1;

int rear=-1;

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

//description: check if queue is full

//input: none

//output: boolean result if queue is full(int)

int queue\_full(){

if(size<=(rear-front)&&(front<rear)){//if queue size exceeded, return true(1)

return 1;

}else{//else, return false(0)

return 0;

}

return 1;

}

//description: check if queue is empty

//input: none

//output: boolean result if queue is full(int)

int queue\_empty(){

if(front==rear){//if stack is front and rear point is same, return true(1)

return 1;

}else{//else, return false(0)

return 0;

}

}

//description: append input data to rear of queue

//input: data to append(int)

//output: none

void push(int num){

if(!queue\_full()){

rear++;

queue[rear]=num;

}else{

printf("Queue is Full!!");

}

}

//description: removes data at top of queue

//input: none

//output: removed data value at top of queue(int)

int pop(){

if(!queue\_empty()){

int atFront=queue[front];

queue[front]=0;

front++;

return atFront;

}else{

printf("Stack is Empty!!");

return 0;

}

}

//description: print numbers in queue in a row

//input: none

//output: data values in size of queue

void printQueue(){

int i;

if(queue\_empty()){

printf("NULL\n");

}

else{

for(i=front; i<=rear; i++){

printf("%d ->",queue[i]);

}

printf("\n");

}

}

//description: test queue functions until queue size be full

//input: none

//output: none

void main(){

printf("Initial Queue: ");

printQueue();//result: NULL

while(!queue\_full()){

int tmp=0;

printf("put number: ");

scanf("%d",&tmp);

printf("\n");

if(0<tmp&&tmp<10){//enqueue tmp at rear into queue if tmp between 1 to 9

push(tmp);

}else{//dequeue at front from queue if tmp is out of the range

pop();

}

printQueue();

}

printf("Queue size is full, end program.");

getchar();

}

**Result)**

