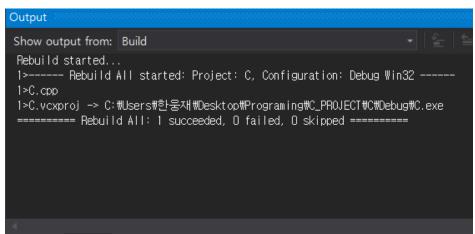
문제해결기법(13967005) 202135592 한웅재 소프트웨어

제출일: 2021. 11. 26

```
Q1. Lab 2 Queue (p. 55)
#define _CRT_SECURE_NO_WARNINGS// or scanf_s
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
#include <stdbool.h>
struct QUEUE {
   int *buf;
   int size;
   int front;
   int back;
}Queue;//global variable
bool Queue_full();
bool Queue empty();
void create_Queue(const int size)
   Queue.buf = (int*)malloc(size * sizeof(int));
   Queue.size = size;
   Queue.front = -1;
   Queue.back = -1;
void release_Queue()
   free(Queue.buf);
   Queue.size = 0;
   Queue.front = -1;
   Queue.back = -1;
void enqueue(const int val) {
   if (Queue_full())
       printf("enqueue() : Queue is full!\n");
       return;
   if (Queue.back==-1)
   {
       Queue.front++;
   Queue.back++;
   Queue.buf[Queue.back] = val;
   printf("enqueue[%d] : %d\n", Queue.back,val);
int dequeue() {
   if (Queue_empty())
   {
       printf("dequeue() : Queue is empty!!\n");
   }
   int val = Queue.buf[Queue.front];
   printf("dequeue[%d] : %d\n", Queue.front, val);
   Queue.front++;
```

```
return val;
bool Queue_full() {
   return Queue.back >= Queue.size - 1;
bool Queue_empty() {
   return Queue.front> Queue.back;
}
int main()
   create_Queue(10);
   int val = 10;
   while (!Queue_full())
       enqueue(val);
       val += 10;
   while (!Queue_empty())
       val = dequeue();
   release_Queue();
   return 0;
}
```



```
enqueue [ 0 ] : 10
enqueue [ 1 ] : 20
enqueue [ 2 ] : 30
enqueue [ 3 ] : 40
enqueue [ 4 ] : 50
enqueue [ 5 ] : 60
enqueue [ 5 ] : 60
enqueue [ 6 ] : 70
enqueue [ 8 ] : 90
enqueue [ 8 ] : 90
enqueue [ 9 ] : 100
dequeue [ 0 ] : 10
dequeue [ 1 ] : 20
dequeue [ 1 ] : 20
dequeue [ 2 ] : 30
dequeue [ 3 ] : 40
dequeue [ 7 ] : 80
enqueue [ 8 ] : 90
enqueue [ 8 ] : 90
enqueue [ 9 ] : 100
dequeue [ 1 ] : 20
dequeue [ 1 ] : 80
dequeue [ 1 ] : 80
dequeue [ 2 ] : 30
dequeue [ 3 ] : 40
dequeue [ 6 ] : 70
dequeue [ 7 ] : 80
dequeue [ 7
```