1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

// ===== 원형큐 코드 시작 ======

#define MAX\_QUEUE\_SIZE 5

typedef int element;

typedef struct { // 큐 타입

element data[MAX\_QUEUE\_SIZE];

int front, rear;

} QueueType;

// 오류 함수

void error(char\* message)

{

fprintf(stderr, "%s\n", message);

exit(1);

}

// 공백 상태 검출 함수

void init\_queue(QueueType\* q)

{

q->front = q->rear = 0;

}

// 공백 상태 검출 함수

int is\_empty(QueueType\* q)

{

return (q->front == q->rear);

}

// 포화 상태 검출 함수

int is\_full(QueueType\* q)

{

return ((q->rear + 1) % MAX\_QUEUE\_SIZE == q->front);

}

// 원형큐 출력 함수

void queue\_print(QueueType\* q)

{

printf("QUEUE(front=%d rear=%d) = ", q->front, q->rear);

if (!is\_empty(q)) {

int i = q->front;

do {

i = (i + 1) % (MAX\_QUEUE\_SIZE);

printf("%d | ", q->data[i]);

if (i == q->rear)

break;

} while (i != q->front);

}

printf("\n");

}

// 삽입 함수

void enqueue(QueueType\* q, element item)

{

if (is\_full(q))

error("큐가 포화상태입니다");

q->rear = (q->rear + 1) % MAX\_QUEUE\_SIZE;

q->data[q->rear] = item;

}

// 삭제 함수

element dequeue(QueueType\* q)

{

if (is\_empty(q))

error("큐가 공백상태입니다");

q->front = (q->front + 1) % MAX\_QUEUE\_SIZE;

return q->data[q->front];

}

// 삭제 함수

element peek(QueueType\* q)

{

if (is\_empty(q))

error("큐가 공백상태입니다");

return q->data[(q->front + 1) % MAX\_QUEUE\_SIZE];

}

// ===== 원형큐 코드 끝 ======

int main(void)

{

int menu\_num;

QueueType queue;

int element;

init\_queue(&queue);

while (1) {

printf("============== menu ==============\n");

printf("1. Input data and Enqueue\n");

printf("2. Dequeue and Print data\n");

printf("3. Print Queue\n");

printf("4. Exit\n");

printf("Select Number : ");

scanf\_s(" %d", &menu\_num);

if (menu\_num == 1)

{

printf("Input the data : ");

scanf\_s(" %d", &element);

enqueue(&queue, element);

}

else if (menu\_num == 2)

{

element = dequeue(&queue);

printf("Dequeue : %d\n", element);

}

else if (menu\_num == 3)

{

queue\_print(&queue);

}

else {

exit(1);

return 0;

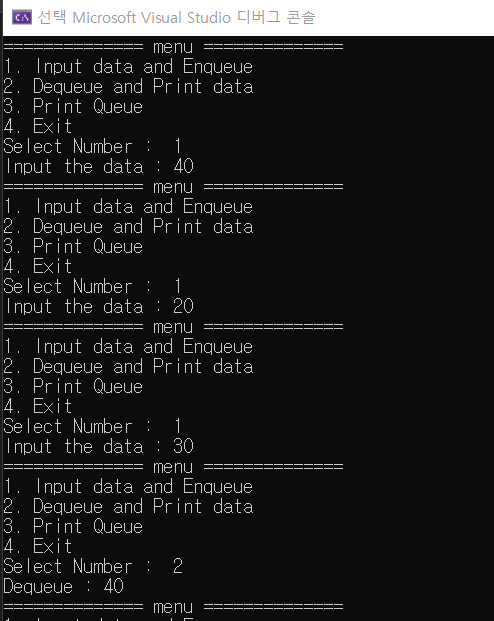
}

}

}

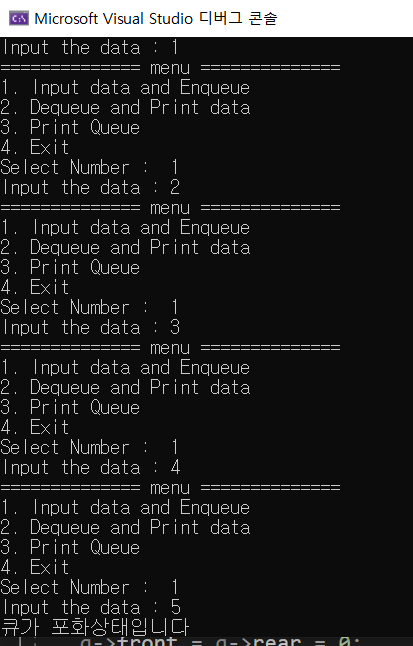
2. 실행 화면

40 20 30 넣고 다 빼기 – 큐 공백

텍스트이(가) 표시된 사진

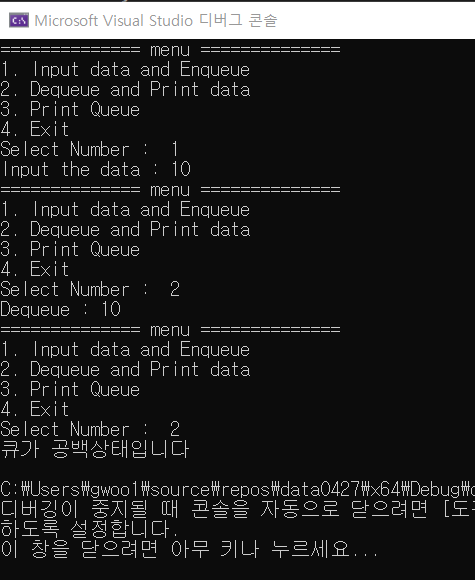
자동 생성된 설명

1 2 3 4 5 넣고 포화 10 20 30 40 50 넣고 포화

도표이(가) 표시된 사진

자동 생성된 설명

10 넣고 다 다빼기 , 10넣은 후 출력 후 종료

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 느낀점

처음에 지난번 시간과 다르게 이번에는 front와 end를 사용하여 선입선출을 하는 큐를 배웠다. 처음 배웠던 큐에서 원형 큐라는 간단한 아이디어로 알고리즘이 이렇게까지 발전되는 과정이 흥미로웠다. 반복문을 사용하여 만드니 숫자들을 저장하고 찾는 하나의 프로그램을 만든 기분이다. 조금만 더 활용한다면 입력하는 숫자를 번호표로 두고 번호표 프로그램으로도 이용할 수 있을 것 같다.