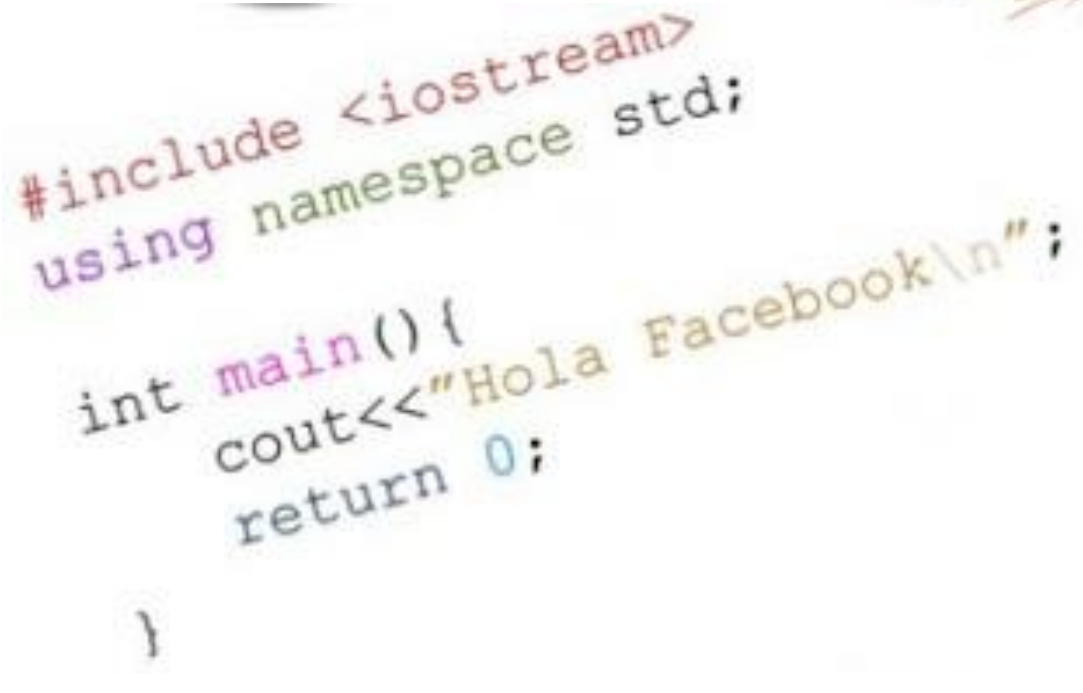
스크래치

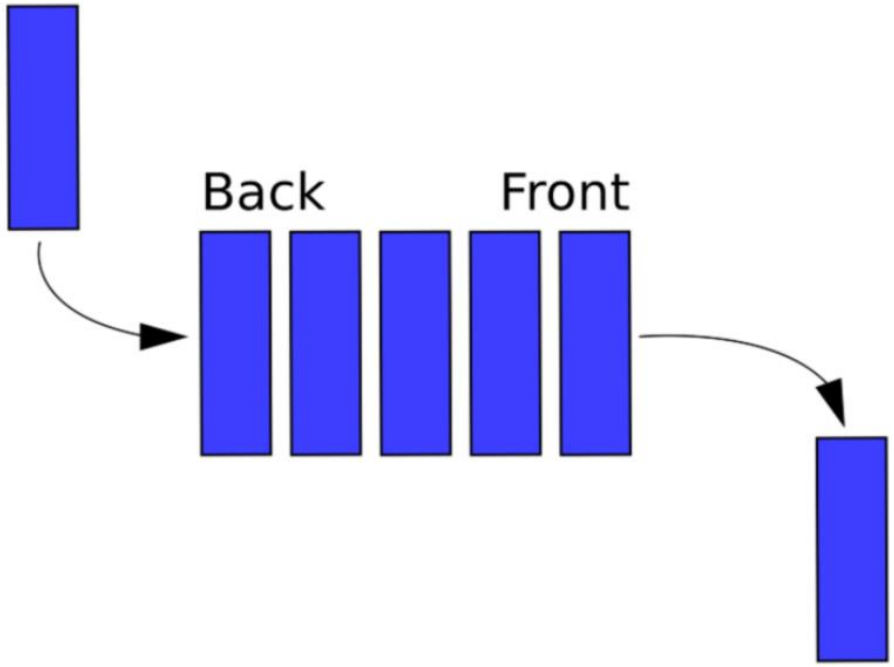
추가마당

- 큐 -


#include <iostream>
using namespace std;
int main(){
cout<<"Hola Facebook\n";
return 0;
}

1. 큐(Queue)

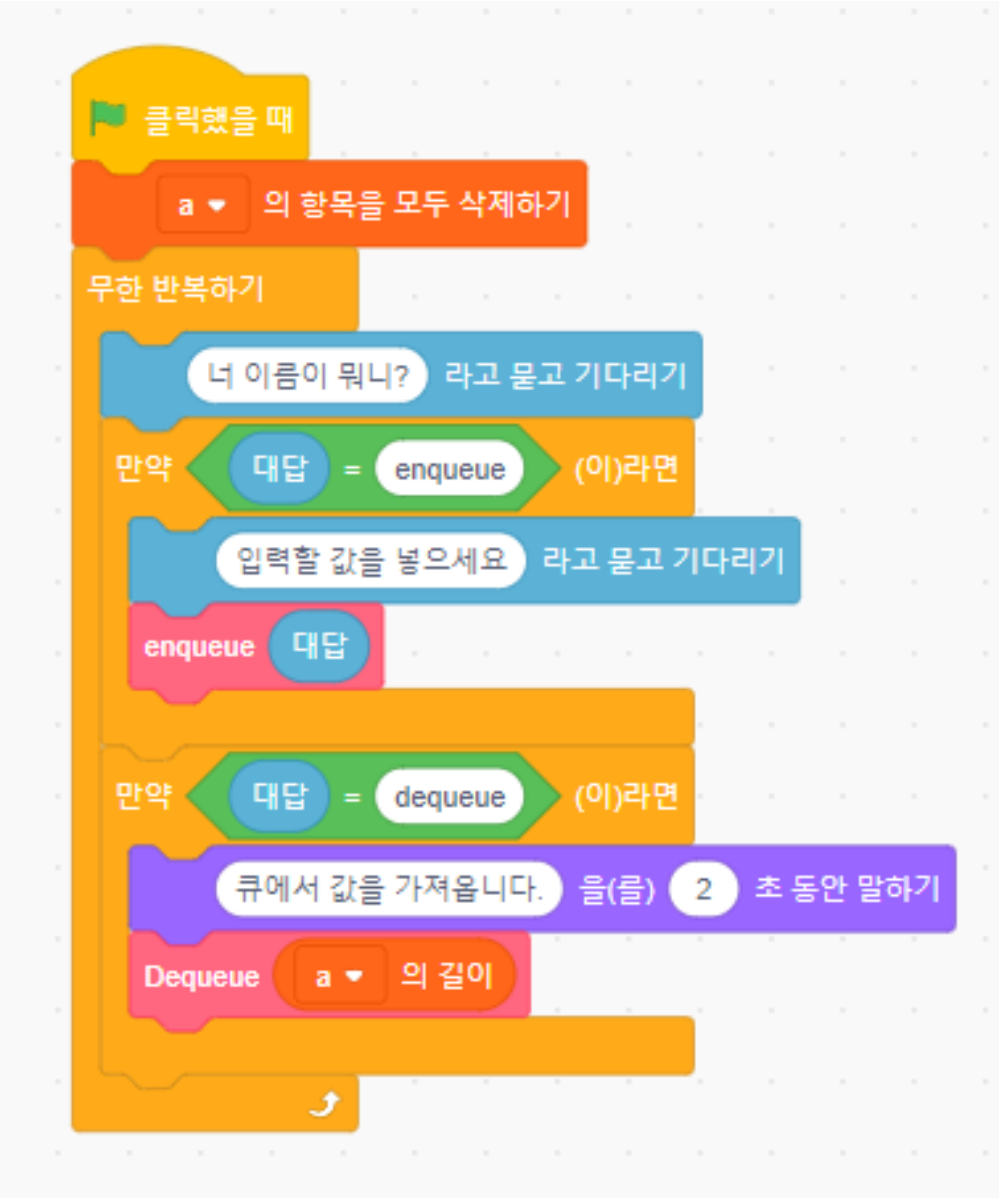
- 큐는 먼저 입력된 데이터가 먼저 출력되는 FIFO(First In First Out) 구조로 저장
  - 스택은 먼저 입력된 데이터가 나중에 출력되는 LIFO(Last In First Out) 구조
  - 데이터 출력은 Front에서 담당, 삽입은 Rear에서 담당
  - 대기열을 지어 기다리고 있다는 개념과도 같음



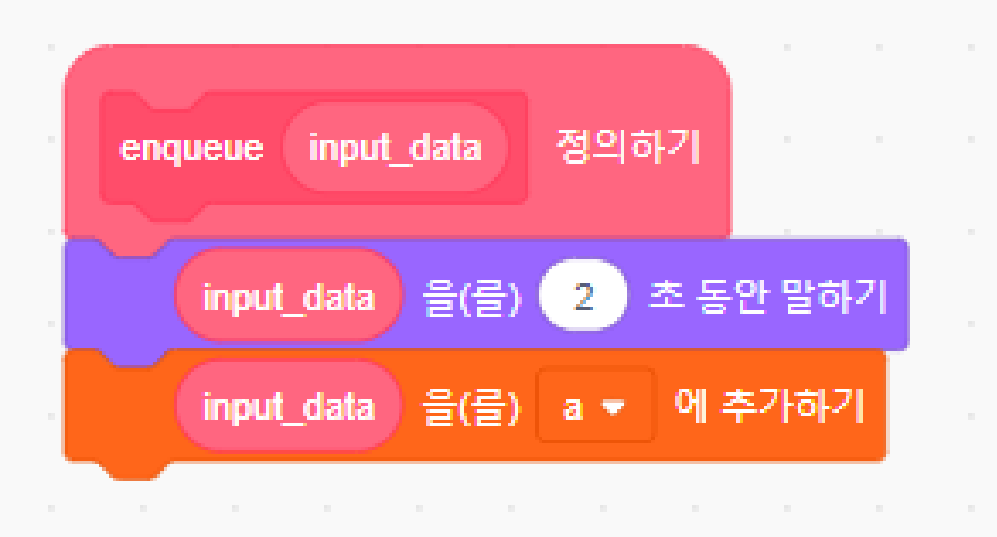
큐 구현에 필요한 것

- 스크래치 리스트 특징을 잘 이용해야 함
- 스크래치에 리스트는 신규로 추가되는 데이터가 기존 데이터에 뒤에 입력되기 때문에 따로 인덱스 작업을 할 필요는 없음
- 블록은 간단하지만 enqueue의 원리는 간단하지 않으므로 인덱스 위치를 잘 조절해주어야 함
- 중간 데이터가 삭제되어도 자동적으로 뒤에 값들이 채워짐

구현하기


클릭했을 때
a 의 항목을 모두 삭제하기
무한 반복하기
너 이름이 뭐니? 라고 묻고 기다리기
만약 대답 = enqueue (이)라면
입력할 값을 넣으세요 라고 묻고 기다리기
enqueue 대답
만약 대답 = dequeue (이)라면
큐에서 값을 가져옵니다. 을(를) 2 초 동안 말하기
Dequeue a 의 길이

Enqueue


enqueue input_data 정의하기
input_data 을(를) 2 초 동안 말하기
input_data 을(를) a 에 추가하기

Dequeue


Dequeue queue_size 정의하기
만약 queue_size < 0 또는 queue_size = 0 (이)라면
큐가 비어 있습니다. 을(를) 2 초 동안 말하기
아니면
a 리스트의 1 번째 항목 을(를) 2 초 동안 말하기
1 번째 항목을 a 에서 삭제하기

스크래치

추가마당

– 팩토리얼 함수 재귀함수로 만들기 –

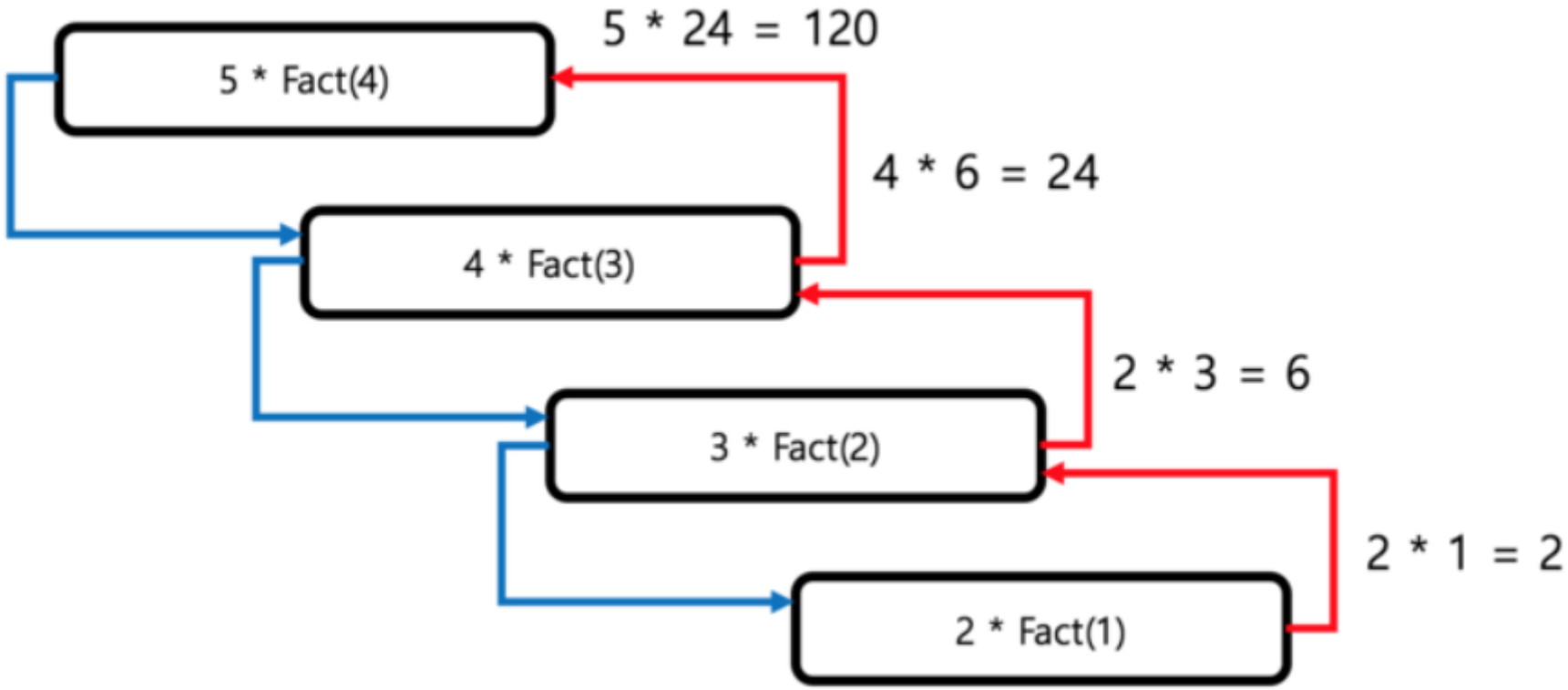
팩토리얼 함수

- 팩토리얼은 1부터 n까지 양의 정수를 차례대로 곱한 값
  - !(느낌표) 기호로 표기
  - 5! = 5 * 4 * 3 * 2 * 1 = 120

- 입력값을 n으로 하였을때, n이 0 또는 1이면 반환은 무조건 1

$$0! = 1 \ (\text{if } n==0)$$
$$1! = 1 \ (\text{if } n==1)$$
$$n * Fact(n-1) \ (\text{else})$$

5! 연산 과정



팩토리얼 함수 구현에 필요한 것

- 스크래치 함수의 단점을 보완해야 함
  - 스크래치는 함수의 반환값을 받을 수 없음
  - 함수에서 반환하고자 하는 값을 전역변수에 저장
  - 그 다음 함수를 호출한 곳에서 그 전역변수에 저장된 값 활용하는 방식!


클릭했을 때
resultlist 의 항목을 모두 삭제하기
구하고 싶은 팩토리얼 값은? 라고 묻고 기다리기
팩토리얼함수 대답
팩토리얼값을 구하기 위한 준비 끝 을(를) 2 초 동안 말하기
fact 을(를) 1 로 정하기
i 을(를) 1 로 정하기
resultlist 의 길이 번 반복하기
fact 을(를) fact * resultlist 리스트의 i 번째 항목 로 정하기
i 을(를) 1 만큼 바꾸기
fact 을(를) 2 초 동안 말하기


팩토리얼함수 n 정의하기
만약 n = 0 (이)라면
1 을(를) resultlist 에 추가하기
만약 n = 1 (이)라면
1 을(를) resultlist 에 추가하기
만약 n = 0 또는 n = 1 이(가) 아니다 (이)라면
n 을(를) resultlist 에 추가하기
팩토리얼함수 n - 1