

과제 2. 스택/큐

과제 2

- ◆ 제출 마감일 : 4월 11일 (일요일) 10:59pm
- ◆ eClass 과제방에 제출
- ◆ 최대 2인까지 팀을 구성할 수 있다(제출은 대표 1인:표지에 팀원 명시).
 - 실습 시험/중간/기말고사에 나올 수 있으므로 모두 내용을 이해해야 한다.
- ◆ 사용 언어
 - C / Java / Python / C++ 중 본인이 선택
 - 본인이 직접 만든 스택과 큐를 이용해야 한다.
- ◆ 제출 양식
 - 보고서 (표지 + 문제 내용과 해결 방안 + 결과 캡처)
 - 소스코드 (**별도 파일로 제출**)
- ◆ 보고서는 Word/HWP/PPT/PDF 중 하나로 제출

타인의 과제를 복사하지 말 것

(1) 계산기

◆ 계산식을 입력하면 결과를 돌려주는 계산기 프로그램을 작성하라.

- 사용되는 연산자는 +, -, *, ^, %, () 이다.
- 입력에 사용되는 수는 100 미만의 정수만 사용한다.
- ^ 는 제곱을 의미하며 3^5 는 3^5 이다. ^의 우선순위는 ()보다 느리고 * 보다 빠르다. 수식에서 ^는 최대 1회만 사용한다.
- %는 나머지를 구하는 연산으로, *와 동일한 우선순위다.
- 수식에서 괄호는 최대 1회만 사용한다. 겹쳐진 괄호는 쓰지 않는다.
- 잘못된 수식이라면 어느 위치에 문제가 있는지 알려줘야 한다.

◆ 실행 예)

$$2+3*4^2-1$$

$$= 49$$

(1) 계산기

◆ 실행 예

$$2+(3*4)^2-12$$

$$=134$$

$$2+(3*4^2-12$$

^ 이 위치에 오류가 있습니다.

$$(11+3)*2^3-12$$

$$=100$$

$$2+(3+*4^2-12$$

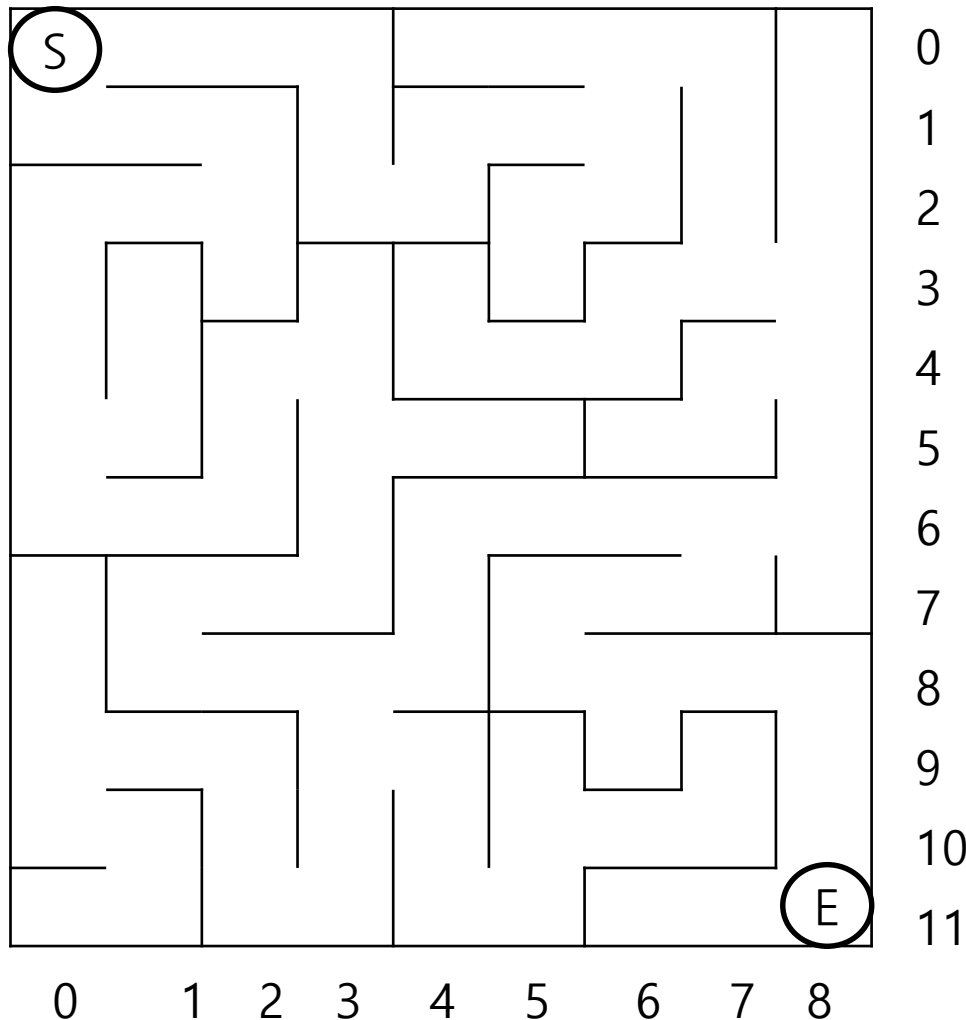
^ 이 위치에 오류가 있습니다.

(2) 미로 탐색

◆ 최적의 미로를 찾는 프로그램을 작성하라.

- 미로는 다음 페이지에 주어져 있다. 이 미로를 사용해야 한다.
- 미로의 길을 찾을 때에는 자신이 만든 스택을 이용해야 한다.
- 스택을 이용할 때마다 스택에서 어떤 정보를 PUSH하고 POP 하는지 화면에 표시해야 한다.
- 길은 여러 개가 있을 수 있다. 그 중 최종적으로 가장 빠른 길을 화면에 표시해야 한다.
 - 거리도 함께 표시한다.
 - 미로 표시는 예쁘게 표시할 필요 없다. 알아볼 수 있는 수준으로만 표시한다. (+, -, | 활용)
- 길 찾기에 앞서 미로를 어떻게 저장할 것인지부터 생각해야 한다.

(2) 미로 탐색



스택 사용 결과는 아래의 예시와 다를 수 있음

```
PUSH(0,0)
PUSH(2,2)
PUSH(5,0)
PUSH(4,3)
PUSH(5,3)
PUSH(3,8)
PUSH(3,9)
PUSH(0,9)
POP(0,9)
POP(3,9)
PUSH(5,10)
PUSH(6,8)
...
```

push, pop 횟수

(이하 생략)

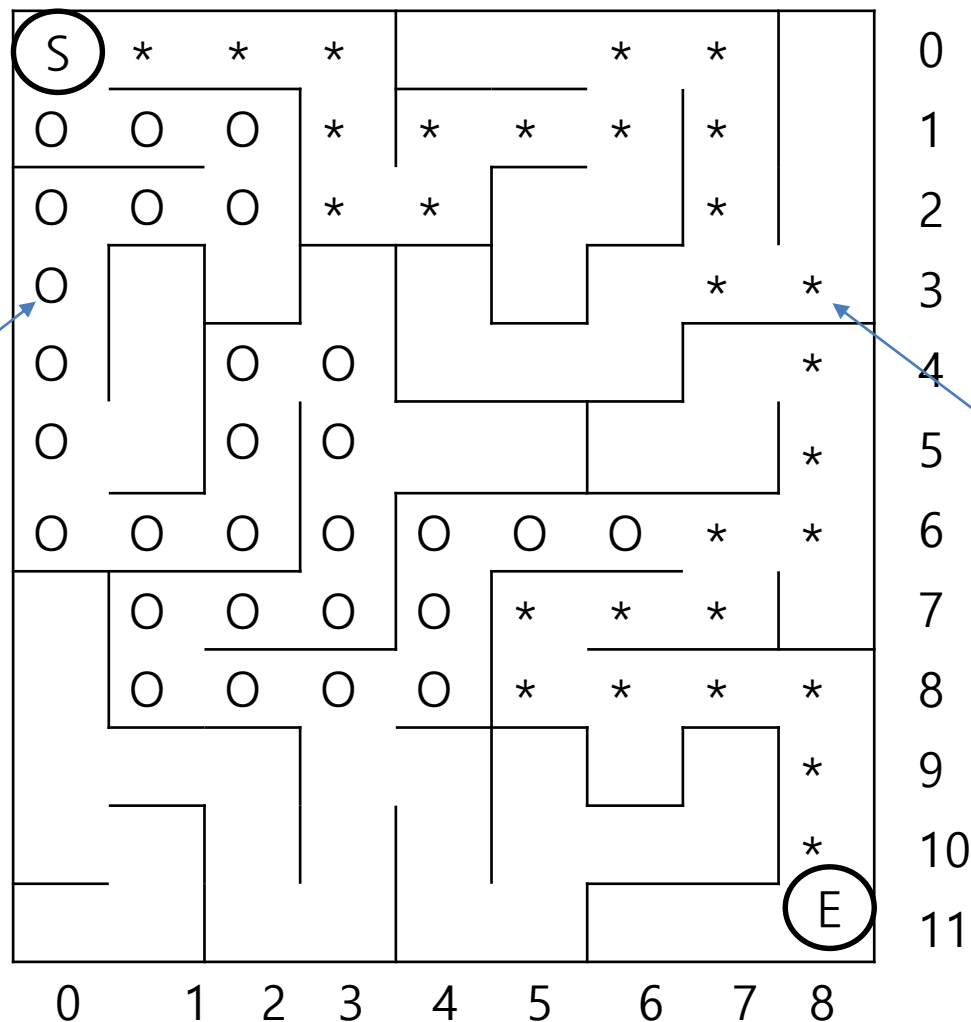
스택 활용 횟수 ##회

모두 2개의 길을 찾았습니다.
가장 짧은 길을 표시합니다
(거리 29).

(2) 미로 탐색

결과는 경로 1개만 표시

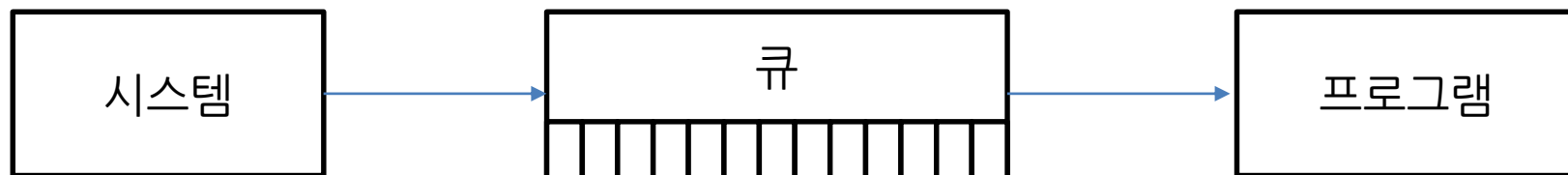
거리:39
멀어서 선택 안한다
따라서 표시도
안한다.



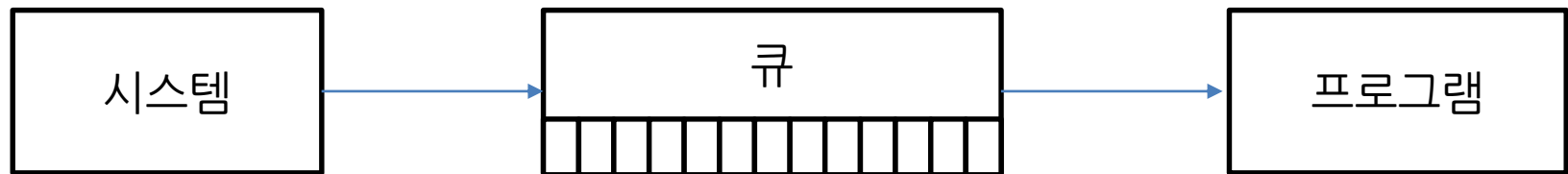
거리 : 29
짧은 길이니
표시한다.

(3) 큐 운영하기

- ◆ 1초에 1개의 문자가 입력되어 큐에 쌓이는 시스템에서 필요한 수만큼 문자를 가져오는 프로그램
 - 우리 시스템은 1초에 1개의 문자가 큐로 들어온다.
 - 사용자가 필요한 데이터의 개수를 입력하면 큐에서 해당 데이터를 가져온다.
 - 입력 큐는 20 글자 크기를 갖는다. 입력 큐의 범위를 초과하면 받아들일 수 없다(에러 표시).
 - **원형 큐**로 동작한다. 직접 원형 큐를 만들어야 한다.



(3) 큐 운영하기



◆ 시스템의 큐 입력

- 1초마다 A~Z의 문자를 순차적으로 입력한다.
 - 0초 : A를 큐에 넣는다. 1초 : B를 큐에 넣는다. 2초 : C를 큐에 넣는다...

◆ 사용자의 동작

- 숫자(1~9)를 넣으면 큐에서 해당 수만큼 가져와 화면에 표시한다.
- 0을 넣으면 큐에서 가져오지 않고 큐의 상태를 화면에 표시한다.

주의할 점 : 1초마다 큐에 넣는 작업을 할 수 없으니 사용자의 입력이 있을 때 몇 초가 지났는지 측정하여 그 수만큼 큐에 넣는 작업을 한꺼번에 진행하자.

(3) 큐 운영하기

◆ 실행 예)

시스템이 시작됩니다.

>>> 0

(SYSTEM) ADDQUEUE(A) F=0 R=1

(SYSTEM) ADDQUEUE(B) F=0 R=2

(SYSTEM) ADDQUEUE(C) F=0 R=3

QUEUE = ABC(3)

>>> 3

(SYSTEM) ADDQUEUE(D) F=0 R=4

(SYSTEM) ADDQUEUE(E) F=0 R=5

DELETEQUEUE() = A, F=1 R=5

DELETEQUEUE() = B, F=2 R=5

DELETEQUEUE() = C, F=3 R=5

RESULT = ABC

QUEUE=DE (2)

>>> 5

(SYSTEM) ADDQUEUE(F) F=3 R=6

(SYSTEM) ADDQUEUE(G) F=3 R=7

DELETEQUEUE() = D, F=4 R=7

DELETEQUEUE() = E, F=5 R=7

DELETEQUEUE() = F, F=6 R=7

DELETEQUEUE() = G, F=7 R=7

DELETEQUEUE() FAIL. QueueEmpty

RESULT = DEFG

QUEUE = (0)

>>>

큐에 입력되는 문자

큐의 상태 보기

3초가 흘러서 큐에 3개가 입력됨

큐의 상태 표시

그 사이에 2초가 흘러서 큐에 2개가 입력됨

3개 가져옴

가져온 결과

큐의 상태 표시

그 사이에 2초가 흘러서 큐에 2개가 입력됨

큐에 데이터가 없어 오류

큐 비어있음