



CHAPTER 7

숫자 추측 게임기



학습 목표

기본 학습 목표

- break 문을 적절하게 사용할 수 있다.
- random 모듈을 이용할 수 있다.
- randint() 함수를 이용하여 임의의 정수를 생성할 수 있다.

심화 학습 목표

- 순차 탐색과 이진 탐색의 원리와 차이를 이해할 수 있다.
- 이진 탐색 알고리즘과 프로그램을 작성할 수 있다.



핵심 학습 요소



핵심 학습 요소

- break 문, random 모듈, randint() 함수



문제 상황



문제 상황

CT

숫자 맞추기 게임 프로그램을 개발하고 싶다. 게임은 다음과 같이 진행된다. 컴퓨터는 1부터 30사이의 숫자 중에서 임의의 숫자를 하나 생성한 후 이를 플레이어에게 맞춰서 보라고 한다. 플레이어는 정해진 횟수 안에 이를 맞히면 이기고 그렇지 않으면 진다. 플레이어의 추측한 숫자가 자신이 가지고 있는 숫자와 같으면 맞았다고 알려주고, 틀리면 자신이 가지고 있는 숫자보다 큰지 작은지를 알려준다. 제한 횟수를 줄이거나 늘려 컴퓨터나 플레이어가 어느 한 쪽이 일방적으로 이기지 못하도록 한다.



문제 분석

✧ 입력

- 추측한 숫자 입력
- 게임 지속 여부를 위해 'YES' 또는 'NO' 입력

✧ 출력

- 숫자 맞힘 여부
- 추측한 숫자가 큰지 작은지



문제 분석

✧ 문제 분해

- 임의의 숫자를 난수로 발생시키기
- 추측한 숫자가 큰지 작은지 알려주기
- 맞힘 여부를 출력해주기
- 제한 횟수 증가 또는 감소시키기

✧ (게임) 규칙

- 컴퓨터가 1부터 30사이에 임의의 숫자를 생성함
- 플레이어가 컴퓨터가 생성한 숫자를 추측함
- 플레이어가 추측한 숫자가 틀린 경우 컴퓨터는 생성한 숫자보다 큰지 작은지 알려줌
- 처음 게임 시작 시 제한 횟수는 5로 함
- 제한 횟수 내에 맞으면 제한 횟수를 1감소시키고 틀리면 1증가시킴

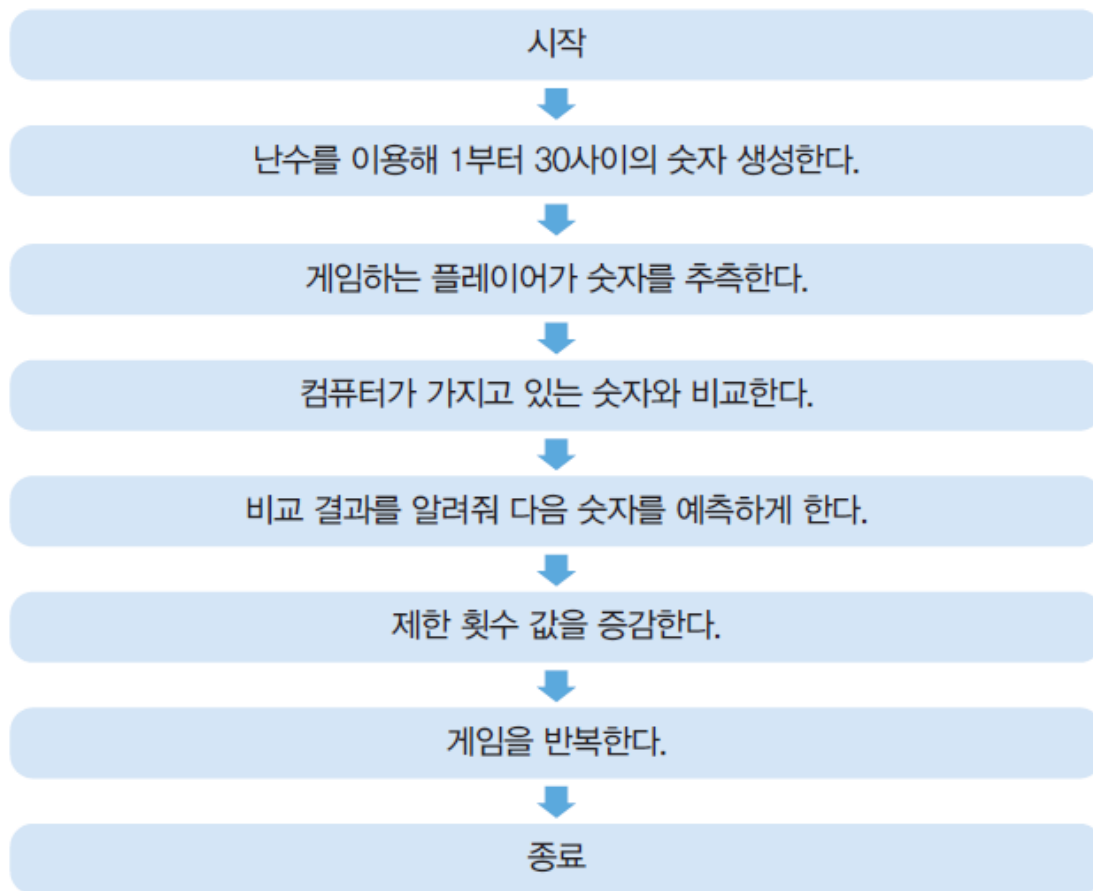
✧ 데이터

- 1부터 30사이의 정수
- 반복 횟수 정수
- 게임 지속 여부 문자열



알고리즘

❖ 전체 알고리즘 뼈대





프로그래밍

❖ 변수

- playerName : 게임하는 사람 이름
- guessNumber: 난수로 생성된 숫자
- limit: 제한 횟수
- count: 추측 횟수
- playAgain: 게임 지속 여부



프로그래밍

- 게임 하는 사람의 입력 - input() 함수
- 임의의 수 생성 - randint() 함수

```
import random #난수를 사용하기 위해 필요함
```

```
print("나랑 숫자 맞추기 게임을 해 보자\n");
```

```
print("네 이름은 뭐니?\n");
```

```
playerName = input()
```

```
count=1
```

```
guessNumber=-1
```

```
limit = 5
```

```
playAgain = 'YES'
```

```
ansNumber = random.randint(1,30)
```

```
print("반가워. " +playerName+ ", 내가 1부터 30사이에 수를 가지고 있어. 맞춰봐\n")
```



실습해보기

실습해보기 7-1

10부터 100까지 임의의 수를 생성하는 프로그램을 작성해 보자.



프로그래밍

■ 두수를 비교 분석 – [if ~ elif ~ else ~]

```
if guessNumber == ansNumber:
    break
elif guessNumber < ansNumber:
    print("추측한 숫자가 컴퓨터가 가지고 있는 숫자보다 작아요")
else:
    print("추측한 숫자가 컴퓨터가 가지고 있는 숫자보다 커요")
count = count+1
```



실습해보기

실습해보기 7-2

위의 if-elif-else 문에서 두 수를 비교하는 순서를 바꿔도 동일한 수행 결과가 나오도록 수정해 보자.

프로그래밍

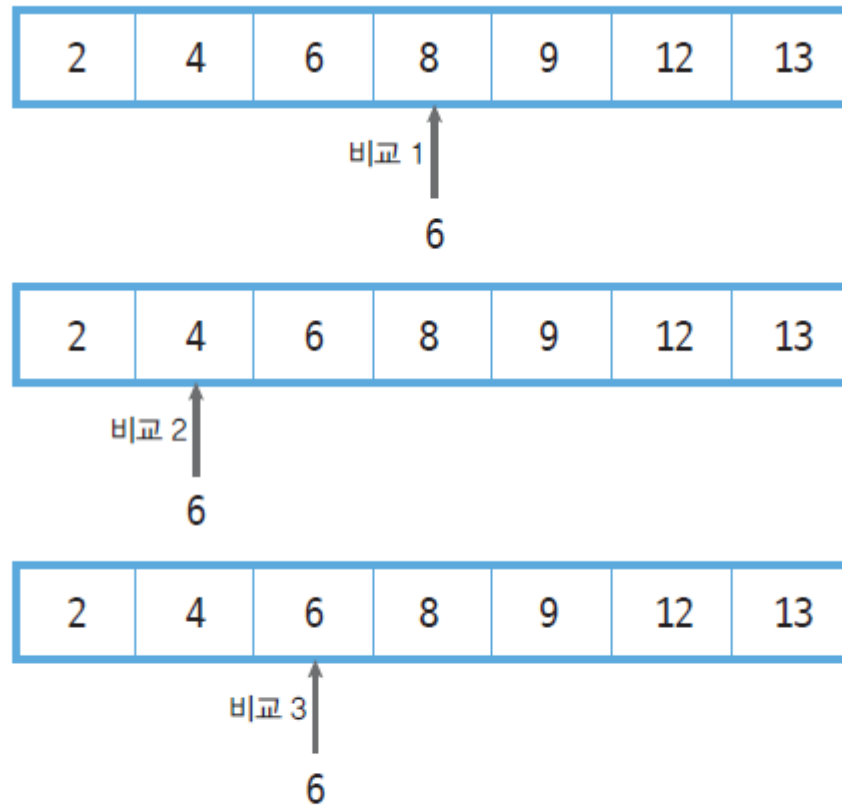
■ 순차 탐색 알고리즘



[그림 7-1] 순차 탐색 기법

프로그래밍

■ 이진 탐색 알고리즘



[그림 7-2] 이진 탐색 기법



문고 답하기

Q 문고 답하기

문 1부터 30 사이에 숫자를 추측하여 맞히는 데 이진 탐색을 한다면 몇 번만에 반드시 맞힐 수 있을까?

Q 문고 답하기

문 1부터 30 사이에 숫자를 추측하여 맞히는 데 순차 탐색을 한다면 최악의 경우 몇 번 만에 맞힐 수 있을까?



프로그래밍

- 숫자를 제한된 횟수 안에 맞힘 여부와 제한 횟수 변경

```
if guessNumber == ansNumber:
    print(count, "번만에 맞혔어요!! 축하해요\n")
    if limit > 1:
        limit = limit - 1
else:
    print("컴퓨터가 가진수는", ansNumber, "야.")
    limit = limit + 1
```




실습해보기

실습해보기 7-3

위의 프로그램을 != 연산자를 이용하여 동일한 기능을 하도록 프로그램을 수정해보자.



프로그래밍 - 전체 프로그램

```
01  import random                #난수를 사용하기 위해 필요함
02
03  print("나랑 숫자 추측 게임을 해 보자\n");
04  print("네 이름은 뭐니?\n");
05  playerName = input()         #플레이어 이름 입력
06  count=1
07  guessNumber=-1
08  limit = 5                     # 제한 횟수 5로 초기화
09  playAgain = 'YES'            # 게임 반복 상태 설정
10
11  ansNumber = random.randint(1,30)
12  print("반가워. " +playerName+ ", 내가 1부터 30사이에 수를 가지고 있어.
13          맞혀봐\n")
14
```



프로그래밍 – 전체 프로그램

```
15 while playAgain == 'YES':
16     print(limit, "번만에 맞혀야 돼")
17     while count <= limit and guessNumber != ansNumber:
18         guessNumber = int(input("추측한 숫자를 입력하세요->"))
19         if guessNumber == ansNumber:
20             break          # 정답을 맞췄으면 안쪽 while 문을 빠져나감
21         elif guessNumber < ansNumber:
22             print("추측한 숫자가 컴퓨터가 가지고 있는 숫자보다 작아요")
23         else:
24             print("추측한 숫자가 컴퓨터가 가지고 있는 숫자보다 커요")
25         count = count+1
26
27     if guessNumber == ansNumber:
28         print(count, "번만에 맞혔어요!! 축하해요\n")
29         if limit > 1:
30             limit = limit -1
31     else:
32         print("컴퓨터가 가진수는", ansNumber, "야.")
33         limit = limit +1
34
35     playAgain = input("게임을 다시 할까요?(YES or NO)\n")
36     count=1
37     guessNumber=-1
```



문고 답하기

Q 문고 답하기

문 이진 탐색 원리를 사용한다면 첫 번째 추측 숫자로 무엇이 좋을까?



테스트와 디버깅

입력값/출력 결과	확인 및 유의사항
나랑 숫자 추측 게임을 해 보자	
네 이름은 뭐니?	
홍길동	
반가워. 홍길동, 내가 1부터 30사이에 수를 가지고 있어. 맞혀봐	
5 번만에 맞혀야 돼	
추측한 숫자를 입력하세요->16	
추측한 숫자가 컴퓨터가 가지고 있는 숫자보다 커요	
추측한 숫자를 입력하세요->8	
추측한 숫자가 컴퓨터가 가지고 있는 숫자보다 커요	
추측한 숫자를 입력하세요->4	
3 번만에 맞혔어요!! 축하해요	
게임을 다시 할까요?(YES or NO)	

- 예상된 테스트 결과인지 확인한다.
- 제한 횟수 보다 더 많이 반복할 때 limit 값을 출력한다.
- 맞힌 경우 제한 횟수가 감소하는 지 확인한다.



심화 활동



심화 활동

CT

게임에서 제한 횟수 내에 정답을 맞히려면 추측하는 수를 이진 탐색을 통해 추측하는 것이 효과적이다. 예를 들면, 이진 탐색이란 1부터 30사이의 숫자를 추측할 때 숫자 15를 추측해보고 맞지 않으면 다시 찾고자 하는 숫자에 가까운 부분만을 대상으로 다시 반으로 나눠 추측하는 것을 반복하는 탐색 기법을 말한다. 이와 같이 이진 탐색을 하면 제일 앞에서부터 순차적으로 탐색하는 순차 탐색보다 얼마나 빨리 찾을 수 있는지를 구체적으로 알아보도록 수정해 보자.