

컴퓨팅적 사고와 문제해결

과제 1

중앙대학교 소프트웨어학부 19학번 유용민
(391040342)

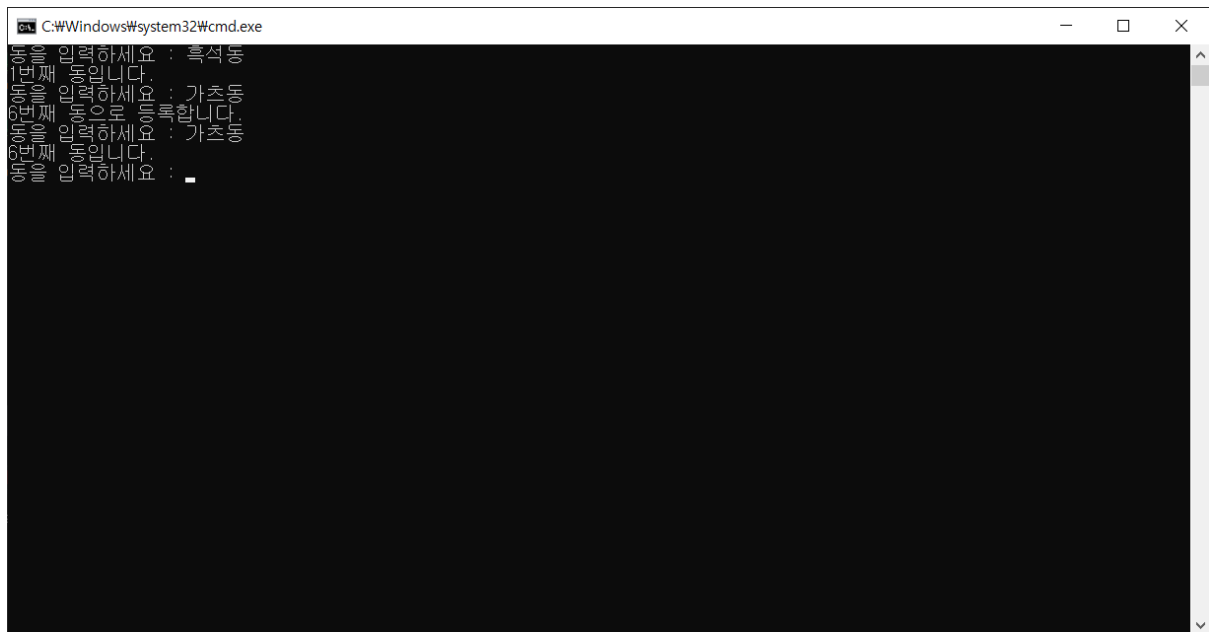
문제 1. 절대값 출력하기

```
1 while 1:
2     inputNum = int(input('숫자를 입력하세요 : '))
3
4     if inputNum == 0:
5         print('프로그램을 종료합니다.')
6         break
7     elif inputNum < 0:
8         # 입력된 값이 음수인 경우, -1을 곱해 그 절대값을 출력
9         print('입력된 숫자의 음수이며, 절대값은 %d입니다.' %(inputNum * -1))
10    elif inputNum > 0:
11        # 입력된 값이 양수인 경우, 그대로 출력
12        print('입력된 숫자는 양수 %d입니다.' %inputNum)
```

if-else 조건문을 통해 양수, 음수의 여부를 파악하고, 음수일 경우 입력된 값에 -1을 곱해 절대값을 구한 뒤 출력하는 코드입니다. 입력값이 0인 경우에는 프로그램을 종료합니다.

문제 2. 리스트에서 데이터 찾기

```
1 listDong = ['흑석동', '사당동', '상도동', '노량진동', '규동']
2
3 while 1:
4     inputDong = str(input('동을 입력하세요 : '))
5     if listDong.count(inputDong):
6         # 입력된 동이 존재하는 경우 몇번째 동인지 출력
7         print('%d번째 동입니다.' %(listDong.index(inputDong)+1))
8     else:
9         # 입력된 동이 존재하지 않는 경우 리스트에 추가(append)
10        listDong.append(inputDong)
11        print('%d번째 동으로 등록합니다.' %(len(listDong)))
```

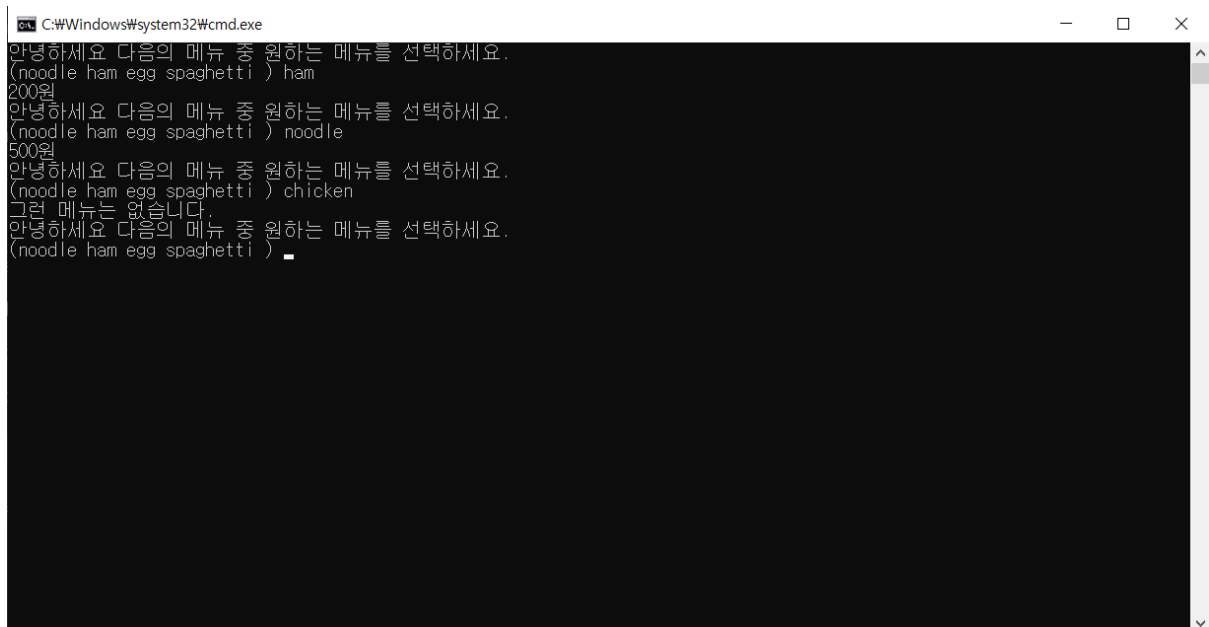


```
C:\Windows\System32\cmd.exe
동을 입력하세요 : 흑석동
1번째 동입니다.
동을 입력하세요 : 가초동
6번째 동으로 등록합니다.
동을 입력하세요 : 가초동
6번째 동입니다.
동을 입력하세요 :
```

주어진 동 요소들을 리스트로 선언합니다. 이후 입력값이 리스트 내에 존재하는 동일 경우 해당 동의 index 값을 출력하고, 존재하지 않는 경우에는 append() 함수를 이용해 리스트의 마지막 요소로 추가해줍니다.

문제 3. 식당 메뉴표

```
1 menu = {'noodle': '500원', 'ham': '200원', 'egg': '100원', 'spaghetti': '900원'}
2
3 # Dictionary로 선언된 메뉴에서 메뉴의 이름(key)만을 추출해 리스트로 생성 및 해당 리스트를 메뉴목록으로 출력하기 위한 String 작업 수행
4 menuList = list(menu.keys())
5 menuStr = '('
6 i = 0
7 for i in range(len(menuList)):
8     menuStr += menuList[i] + ' '
9     i += 1
10 menuStr += ')'
11
12 while 1:
13     print('안녕하세요 다음의 메뉴 중 원하는 메뉴를 선택하세요.')
14     inputMenu = str(input(menuStr+ ' '))
15
16     if inputMenu in menu:
17         # 입력된 메뉴가 존재하는 메뉴인 경우 메뉴 Dictionary로부터 가격을 가져와 출력
18         print(menu.get(inputMenu))
19     else:
20         print('그런 메뉴는 없습니다.')
```

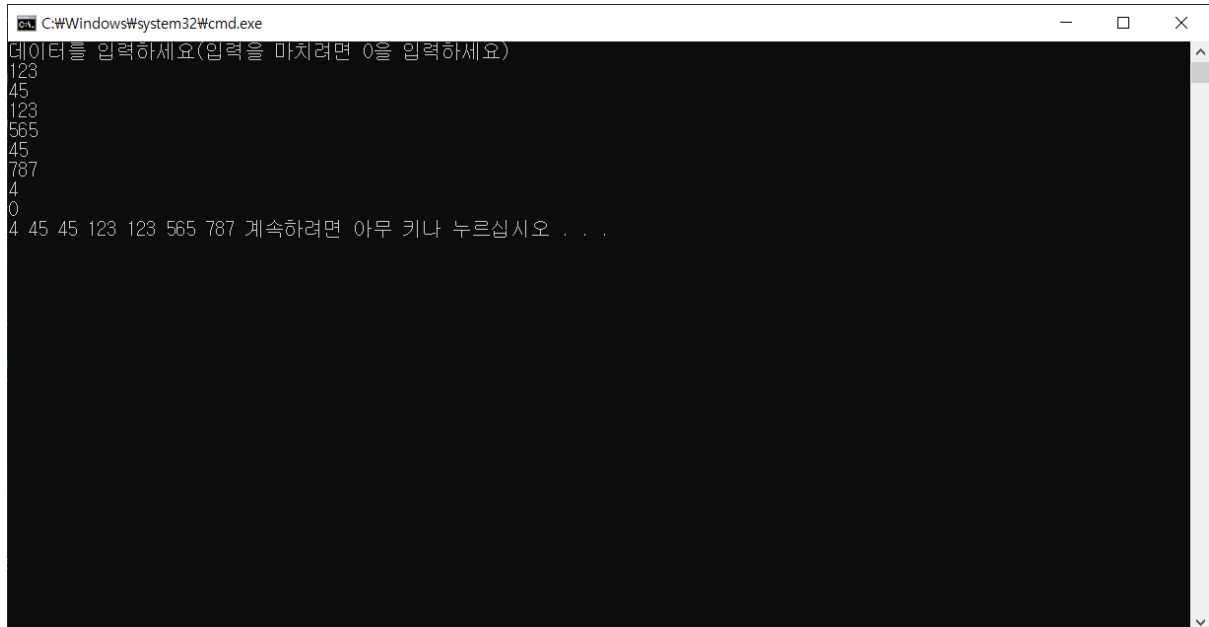


```
C:\Windows\system32\cmd.exe
안녕하세요 다음의 메뉴 중 원하는 메뉴를 선택하세요.
(noodle ham egg spaghetti ) ham
200원
안녕하세요 다음의 메뉴 중 원하는 메뉴를 선택하세요.
(noodle ham egg spaghetti ) noodle
500원
안녕하세요 다음의 메뉴 중 원하는 메뉴를 선택하세요.
(noodle ham egg spaghetti ) chicken
그런 메뉴는 없습니다.
안녕하세요 다음의 메뉴 중 원하는 메뉴를 선택하세요.
(noodle ham egg spaghetti )
```

Dictionary 를 이용해 주어진 메뉴와 가격을 저장해둡니다. 프로그램 실행 시, Dictionary로부터 메뉴만을 추출해 String으로 변환하여 사용자에게 출력해줍니다. 입력받은 메뉴가 Dictionary 에 존재하는 메뉴일 경우, 해당 메뉴의 가격을 불러와 출력하고, 존재하지 않는 경우에는 지정된 문구를 출력합니다.

문제 4. 오름차순 출력

```
1 inputList = []
2 count = 0;
3
4 print('데이터를 입력하세요(입력을 마치려면 0을 입력하세요)')
5 while 1:
6     inputNum = int(input())
7     if inputNum:
8         # 입력값이 0이 아닌경우 값을 리스트에 추가(append)
9         inputList.append(inputNum)
10        count += 1
11    else:
12        break
13
14 # 입력된 값들을 리스트 내에서 크기순으로 정렬
15 inputList.sort()
16
17 for i in range(len(inputList)):
18     print(inputList[i], end=' ')
```

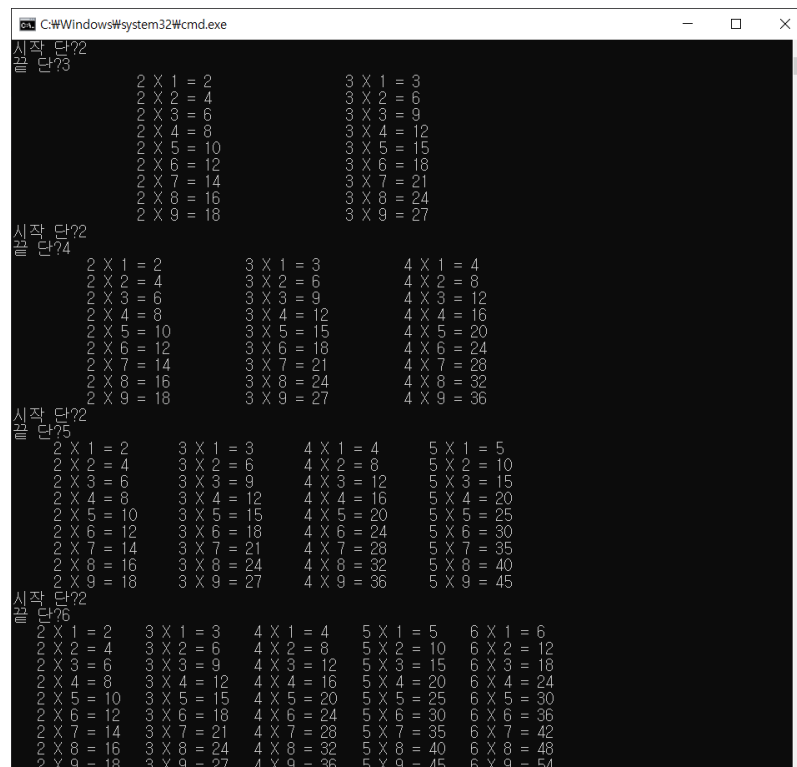


```
C:\Windows\system32\cmd.exe
데이터를 입력하세요(입력을 마치려면 0을 입력하세요)
123
45
123
565
45
787
0
4 45 45 123 123 565 787 계속하려면 아무 키나 누르십시오 . . .
```

while 반복문을 이용해 사용자로 부터 숫자를 입력받아 리스트로 저장하는 과정을 반복합니다. 입력값이 0 인 경우에는 break 함수를 호출해 반복문을 종료한 뒤 리스트를 sort()함수를 이용해 오름차순으로 정렬하여 출력합니다.

문제 5. 구구단

```
1 while 1:
2     start = int(input('시작 단?'))
3     stop = int(input('끝 단?'))
4
5     if stop < start:
6         print('시작 단의 입력값이 끝 단의 입력값보다 작아야 합니다. 다시 입력해주세요.')
7         continue
8
9     ...
10    시작 단과 끝 단의 각 입력값 사이 차이가 4 이하라는 문제의 제시 조건을 만족하기 위한 if문
11    각 입력값 사이 차 크기에 따라 출력시의 간격 크기를 지정
12    ...
13    if (stop - start) > 4:
14        print('시작 단과 끝 단 사이의 차이가 4 이하가 되어야 합니다. 다시 입력해주세요.')
15        continue
16    elif (stop - start) == 4:
17        endStr = ' '
18    elif (stop - start) == 3:
19        endStr = ' '
20    elif (stop - start) == 2:
21        endStr = ' '
22    elif (stop - start) == 1:
23        endStr = ' '
24
25    for i in range(1, 10):
26        for j in range(start, stop + 1):
27            print(endStr, j, 'X', i, '=', j * i, end=' ' if j*i > 9 else ' ')
28            j += 1
29        print(' ')
```



```
C:\Windows\system32\cmd.exe
시작 단?2
끝 단?3
2 X 1 = 2      3 X 1 = 3
2 X 2 = 4      3 X 2 = 6
2 X 3 = 6      3 X 3 = 9
2 X 4 = 8      3 X 4 = 12
2 X 5 = 10     3 X 5 = 15
2 X 6 = 12     3 X 6 = 18
2 X 7 = 14     3 X 7 = 21
2 X 8 = 16     3 X 8 = 24
2 X 9 = 18     3 X 9 = 27

시작 단?2
끝 단?4
2 X 1 = 2      3 X 1 = 3      4 X 1 = 4
2 X 2 = 4      3 X 2 = 6      4 X 2 = 8
2 X 3 = 6      3 X 3 = 9      4 X 3 = 12
2 X 4 = 8      3 X 4 = 12     4 X 4 = 16
2 X 5 = 10     3 X 5 = 15     4 X 5 = 20
2 X 6 = 12     3 X 6 = 18     4 X 6 = 24
2 X 7 = 14     3 X 7 = 21     4 X 7 = 28
2 X 8 = 16     3 X 8 = 24     4 X 8 = 32
2 X 9 = 18     3 X 9 = 27     4 X 9 = 36

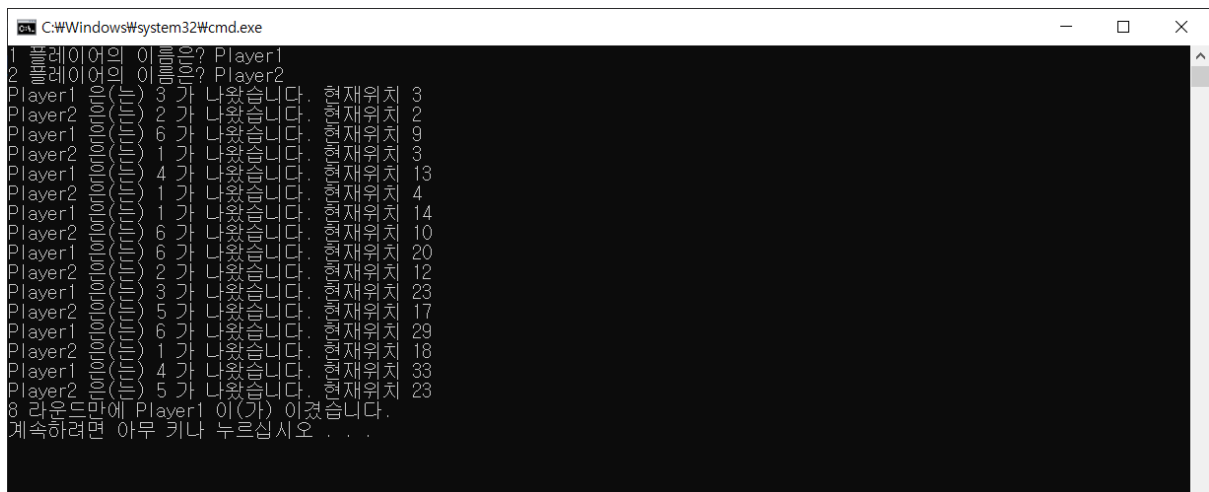
시작 단?2
끝 단?5
2 X 1 = 2      3 X 1 = 3      4 X 1 = 4      5 X 1 = 5
2 X 2 = 4      3 X 2 = 6      4 X 2 = 8      5 X 2 = 10
2 X 3 = 6      3 X 3 = 9      4 X 3 = 12     5 X 3 = 15
2 X 4 = 8      3 X 4 = 12     4 X 4 = 16     5 X 4 = 20
2 X 5 = 10     3 X 5 = 15     4 X 5 = 20     5 X 5 = 25
2 X 6 = 12     3 X 6 = 18     4 X 6 = 24     5 X 6 = 30
2 X 7 = 14     3 X 7 = 21     4 X 7 = 28     5 X 7 = 35
2 X 8 = 16     3 X 8 = 24     4 X 8 = 32     5 X 8 = 40
2 X 9 = 18     3 X 9 = 27     4 X 9 = 36     5 X 9 = 45

시작 단?2
끝 단?6
2 X 1 = 2      3 X 1 = 3      4 X 1 = 4      5 X 1 = 5      6 X 1 = 6
2 X 2 = 4      3 X 2 = 6      4 X 2 = 8      5 X 2 = 10     6 X 2 = 12
2 X 3 = 6      3 X 3 = 9      4 X 3 = 12     5 X 3 = 15     6 X 3 = 18
2 X 4 = 8      3 X 4 = 12     4 X 4 = 16     5 X 4 = 20     6 X 4 = 24
2 X 5 = 10     3 X 5 = 15     4 X 5 = 20     5 X 5 = 25     6 X 5 = 30
2 X 6 = 12     3 X 6 = 18     4 X 6 = 24     5 X 6 = 30     6 X 6 = 36
2 X 7 = 14     3 X 7 = 21     4 X 7 = 28     5 X 7 = 35     6 X 7 = 42
2 X 8 = 16     3 X 8 = 24     4 X 8 = 32     5 X 8 = 40     6 X 8 = 48
2 X 9 = 18     3 X 9 = 27     4 X 9 = 36     5 X 9 = 45     6 X 9 = 54
```

시작단과 끝단의 숫자를 입력받되, 문제에서 지정된 조건에 따라 시작단과 끝 단 사이의 차이가 4 이하가 되어야 합니다. 시작단과 끝단 사이의 차의 크기에 따라 출력 시 간격이 달라지며, 이는 적절한 규칙성을 찾지 못해 String 으로 하드코딩 하였습니다. for 반복문을 이용해 각 단을 1 부터 9 까지 곱하는 작업을 반복 및 출력합니다.

문제 6. 경주 게임

```
1 import random
2
3 player1 = str(input('1 플레이어의 이름은? '))
4 player2 = str(input('2 플레이어의 이름은? '))
5
6 # 현재 진행중인 라운드
7 round = 1
8
9 # 두 플레이어의 위치를 출발지점(0)으로 초기화
10 current1 = 0
11 current2 = 0
12
13 while current1<30 and current2<30:
14     # 플레이어의 주사위 값
15     dice1 = random.randint(1, 6)
16     dice2 = random.randint(1, 6)
17
18     # 각 플레이어의 현재 위치에 주사위 값을 더함
19     current1 += dice1
20     current2 += dice2
21
22     print(player1, '은(는)', dice1, '가 나왔습니다. 현재위치', current1)
23     print(player2, '은(는)', dice2, '가 나왔습니다. 현재위치', current2)
24
25     if current1>=30 and current2<30: # 1 플레이어만 30 이후의 위치에 도달한 경우
26         print(round, '라운드만에 ', player1, '이(가) 이겼습니다.')
27         break
28     elif current1<30 and current2>=30: # 2 플레이어만 30 이후의 위치에 도달한 경우
29         print(round, '라운드만에 ', player2, '이(가) 이겼습니다.')
30         break
31     elif current1>=30 and current2>=30: # 두 플레이어가 동시에 30 이후의 위치에 도달한 경우
32         print(round, '라운드만에 무승부로 게임이 종료되었습니다.')
33         break
34
35 # 게임이 종료되지 않은 경우 현재 라운드에 1을 더하고 반복문 다시 실행
36 round += 1
```

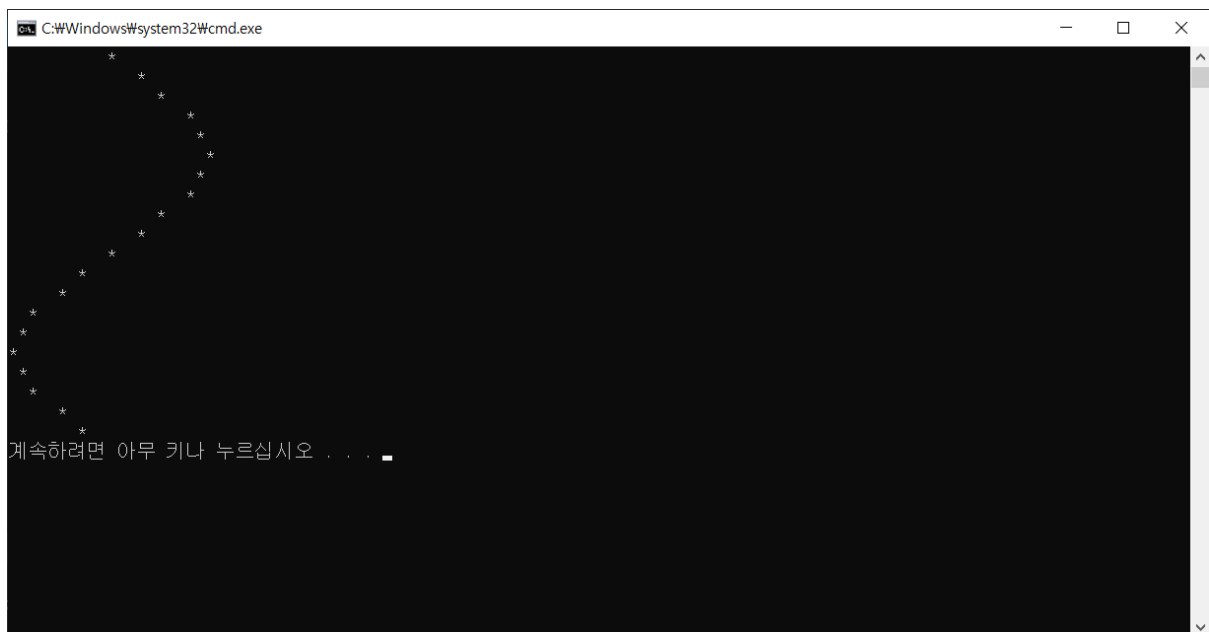


```
C:\Windows\System32\cmd.exe
1 플레이어의 이름은? Player1
2 플레이어의 이름은? Player2
Player1 (1) 3 가 나왔습니다. 현재위치 3
Player2 (1) 2 가 나왔습니다. 현재위치 2
Player1 (2) 6 가 나왔습니다. 현재위치 9
Player2 (2) 1 가 나왔습니다. 현재위치 3
Player1 (3) 4 가 나왔습니다. 현재위치 13
Player2 (3) 1 가 나왔습니다. 현재위치 4
Player1 (4) 1 가 나왔습니다. 현재위치 14
Player2 (4) 6 가 나왔습니다. 현재위치 10
Player1 (5) 6 가 나왔습니다. 현재위치 20
Player2 (5) 2 가 나왔습니다. 현재위치 12
Player1 (6) 3 가 나왔습니다. 현재위치 23
Player2 (6) 5 가 나왔습니다. 현재위치 17
Player1 (7) 6 가 나왔습니다. 현재위치 29
Player2 (7) 1 가 나왔습니다. 현재위치 18
Player1 (8) 4 가 나왔습니다. 현재위치 33
Player2 (8) 5 가 나왔습니다. 현재위치 23
8 라운드만에 Player1 이(가) 이겼습니다.
계속하려면 아무 키나 누르십시오 . . .
```

프로그램 시작 시 먼저 두 플레이어의 이름을 입력 받습니다. 이후 두 플레이어가 각각 주사위를 던지는 작업을 수행한 뒤 주사위 값 만큼 위치를 이동합니다. 위치가 30 이상인 경우 해당 플레이어가 이긴 것으로 처리하며, 동일 회차에 두 플레이어가 모두 30 이상의 위치에 도달한 경우는 무승부로 처리합니다.

문제 7. 사인 그래프 그리기

```
1 from math import sin, pi
2
3 for i in range(20): # 20개의 *을 이용해 sin 함수의 한 주기 그래프를 그리도록 하기 위해 20번 반복
4     # i가 0일 때 sin(0), i가 10일 때 sin(pi), i가 20일 때 sin(2 * pi)의 값을 갖도록 range값 지정 후 해당 값만큼 공백 출력
5     for j in range(10 + int(10 * (sin(i * pi / 10)))):
6         print(' ', end='')
7     # 공백 이후에 1개의 *을 출력해 그래프가 작성되도록 구현
8     print('*')
```



$\sin()$ 함수의 경우 각도를 Degree 단위가 아닌 Radian 단위로 입력해 주어야 합니다. \sin 그래프의 한 주기는 0 부터 2π 까지로 이루어집니다. 20 개의 *을 이용해 그래프를 그리게 되므로, for 반복문에서 i 값이 20 일 때 2π 가 되도록 $\sin()$ 함수의 파라미터 값을 수식으로 지정해줍니다. \sin 함수의 값은 -1 부터 1 까지의 값을 갖게 되므로, 이에 10 을 곱하면 -10 부터 10 이 되고, 이 값에 10 을 더해주면 0 부터 20 사이의 값을 갖습니다. 이 0 부터 20 이라는 값을 개수로 공백이 출력되도록 하고, 이 이후에 *이 출력되도록 하면 \sin 그래프의 한 주기가 출력됩니다.