

3. Loop Statements (반복문) - Answer

Initial Commit: 2020.08.01

Private lesson (OOP Basics with Python, August 2020)

본 답안들은 제시된 문제를 푸는 방법 중 하나일 뿐입니다.

(1) 정수를 입력받아 n 변수에 저장하고, 1부터 n까지의 모든 정수를 더한 값을 반복문을 이용해 얻어 출력하세요.

```
n = int(input('Enter a number: '))
i = 1
result = 0

while i <= n:
    result = result + i
    i = i + 1

print(result)
```

(2-1) 2부터 9까지의 정수 범위에서 n 변수를 입력받아 저장하고, 구구단 n단을 출력하세요. 출력에는 **while**문을 사용해야 합니다.

```
n = int(input('Enter a number: '))
i = 1
while i <= 9:
    print(n, '*', i, '=', n*i)
    i += 1
```

(2-2) 2-1과 동일한 문제입니다. 대신 **for**문을 사용해 프로그래밍하세요.

```
n = int(input('Enter a number: '))
for i in range(1, 10):
    print(n, '*', i, '=', n*i)
```

(3) st 변수에 문자열을 입력받아 저장한 후, **for-in** 문을 사용하여 해당 문자열의 첫 단어만을 출력하세요. (문자열은 영어를 기준으로 하기에, 단어는 띄어쓰기로 구분됩니다.)

```
st = input('Input: ')
for c in st:
    if (c == ' '):
```

```
break
print(c, end='')
```

(4-1) n 변수에 정수를 입력받아 저장하세요. 그 후 입력받은 정수 n을 소인수분해하여 출력하세요. n은 한 자릿수의 소인수만을 가진다고 가정합니다.

```
n = int(input('Enter a number: '))
while n > 1:
    if n % 2 == 0:
        print(2)
        n /= 2
    elif n % 3 == 0:
        print(3)
        n /= 3
    elif n % 5 == 0:
        print(5)
        n /= 5
    elif n % 7 == 0:
        print(7)
        n /= 7
```

(4-2) 4-1번 문제에 대한 추가 도전 과제입니다. 정수 배열 arr = [2, 3, 5, 7]을 선언하고 이중 반복문을 사용하여 동일한 프로그램을 구현하세요.

```
n = int(input('Enter a number: '))
arr = [2, 3, 5, 7]
while n > 1:
    for i in arr:
        if n % i == 0:
            print(i)
            n /= i
            break
```

(5) n 변수에 정수를 입력받아 저장하세요. 아래의 실행 예시 1, 2를 보고 규칙을 찾아 입력받은 n에 윗게 대응하는 패턴을 출력하세요. 출력에는 **while문** 또는 **for문**을 이중으로 사용해야 합니다.

```
n = int(input('Enter a number: '))
for i in range(0, n):
    for j in range(0, i + 1):
        print(n, end='')
    print('')
```

(6) n 변수에 정수를 입력받아 저장하세요. 아래의 실행 예시 1, 2를 보고 규칙을 찾아 입력받은 n에 옳게 대응하는 패턴을 출력하세요. 출력에는 **while문** 또는 **for문**을 이중으로 사용해야 합니다.

```
n = int(input('Enter a number: '))
for i in range(0, n):
    for j in range(0, 2 * n - 1):
        if (j >= (n-1) - i and j <= (n-1) + i):
            print(n, end='')
        else:
            print(0, end='')
    print('')
```

(7-1) 1부터 8까지의 정수가 임의적으로 뒤섞인 배열 변수 arr을 선언하고, 해당 변수를 오름차순으로 정렬하여 출력하세요.

Example: arr = [7, 8, 2, 4, 1, 3, 5, 6]

```
arr = [7, 8, 2, 4, 1, 3, 5, 6]
for i in range(0, 7):
    for j in range(0, 7):
        if (arr[j] > arr[j+1]):
            k = arr[j]
            arr[j] = arr[j+1]
            arr[j+1] = k

for i in range(0, 8):
    print(arr[i])
```

(7-2) 7-1번 문제에 대한 추가 도전 과제입니다. 정렬 알고리즘의 최소 시간복잡도를 추정해보고 그 근거를 생각해 보세요.

- (Answer) Quick Sort - $O(n \log n)$

(Extra) Charlie, Brice, Bill, Ray 네 학생이 Monthly Test를 치게 되었습니다. (학생들의 이름과 점수는 배열 변수에 저장하는 것을 추천합니다.) Monthly Test의 결과를 통해 Student of Month (SOM)을 선정하고자 합니다. 선정 방식은 다음과 같습니다:

- 네 학생 중 SOM 점수가 가장 높은 학생이 SOM에 선정됩니다.
- Monthly Test의 점수 범위는 0 ~ 100점이며, Pass 기준은 80점입니다.
- Pass를 받은 학생은 Retest를 더 이상 보지 않습니다. 또한 Retest는 최대 2번까지 볼 수 있습니다.
- Retest를 치르지 않고 바로 Pass를 받은 학생의 SOM 점수는 Monthly test 점수와 같습니다.
- 첫 번째 Retest를 통과한 학생의 SOM 점수는 Retest 점수의 90%입니다. 소수점 아래는 버립니다.

- 두 번째 Retest를 통과한 학생의 SOM 점수는 2번째 Retest 점수의 80%입니다. 소수점 아래는 버립니다.
- 끝까지 Pass를 받지 못한 학생의 SOM 점수는 세 번의 테스트 점수 중 제일 높은 점수의 70%입니다. 소수점 아래는 버립니다.

모든 학생이 Monthly Test를 Pass하거나 세 번의 테스트를 모두 치를 때까지 학생들의 점수를 반복하여 입력받으세요. 그 후 각 학생의 SOM 점수를 출력한 후, SOM을 선정하세요.

```
names = ['Charlie', 'Brice', 'Bill', 'Ray']
scores = [0, 0, 0, 0]
passes = [False, False, False, False]

for i in range(0, 4):
    score = int(input('[1st] ' + names[i] + ' Score: '))
    if score >= 80:
        passes[i] = True
        scores[i] = score

for i in range(0, 4):
    if passes[i]:
        continue
    score = int(input('[2nd] ' + names[i] + ' Score: '))
    if score >= 80:
        passes[i] = True
        scores[i] = int(score * 0.9)
    elif score > scores[i]:
        scores[i] = score

for i in range(0, 4):
    if passes[i]:
        continue
    score = int(input('[3rd] ' + names[i] + ' Score: '))
    if score >= 80:
        scores[i] = int(score * 0.8)
    else:
        if score > scores[i]:
            scores[i] = score
        scores[i] = int(scores[i] * 0.7)
```

```
highest = 0
som = ''
for i in range(0, 4):
    print(names[i], 'SOM Score:', scores[i])
    if highest < scores[i]:
        highest = scores[i]
        som = names[i]

print('SOM:', som)
```