

1. 중복되는 데이터가 없이 정수를 입력 받아서 리스트로 만든다. 리스트의 원소 중 완전수, 과잉수, 부족수의 합을 구해 s1, s2, s3에 각각 저장한 다음, 이를 반환하는 함수 sumPerfect(a)를 사용하여 프로그램을 작성하라. 2보다 작은 수를 입력하면, '2 이상의 수만 입력하세요.'라는 메시지를 출력한다. 다음 예에서 6은 완전수, 12와 18은 과잉수, 4, 5는 부족수이다.

- 순서도는 sumPerfect(a)에 대해 그린다.

>>>

정수 입력(종료시는 999) : 4

정수 입력(종료시는 999) : 12

정수 입력(종료시는 999) : 1

2 이상의 수만 입력하세요.

정수 입력(종료시는 999) : 18

정수 입력(종료시는 999) : 6

정수 입력(종료시는 999) : 5

정수 입력(종료시는 999) : 4

정수 입력(종료시는 999) : 5

정수 입력(종료시는 999) : 999

생성된 리스트 : [4, 12, 18, 6, 5]

완전수의 합 : 6

과잉수의 합 : 30

부족수의 합 : 9

>>>

과잉수이면 1, 완전수이면 2, 부족수이면 3을 반환하는 함수 isPerfect(a)는 다음과 같다.

```
def isPerfect(x):
```

```
    s = 0
```

```
    for i in range(1, int(x/2)+1):
```

```
        if x % i == 0:
```

```
            s += i
```

```
    if s == x:
```

```
        return 1
```

```
elif s > x:
    return 2
else:
    return 3
```

---

---

```
def isPerfect(x):
    s = 0
    for i in range(1, int(x/2)+1):
        if x % i == 0:
            s += i
    if s == x:
        return 1
    elif s > x:
        return 2
    else:
        return 3
```

```
def sumPerfect(a):
    n = len(a)
    s1, s2, s3 = 0, 0, 0
    for i in range(n):
        if isPerfect(a[i]) == 1:
            s1 += a[i]
        elif isPerfect(a[i]) == 2:
            s2 += a[i]
        else:
            s3 += a[i]
    return (s1, s2, s3)
```

```
data = int(input('정수 입력(종료시는 999) : '))
nums = []
while data != 999:
    if data <= 1:
        print('2 이상의 수만 입력하세요.')
    elif nums.count(data) == 0:
        nums.append(data)
```

```
data = int(input('정수 입력(종료시는 999) : '))
print('생성된 리스트 :', nums)
result = sumPerfect(nums)
print()
print('완전수의 합 : ', result[0])
print('과잉수의 합 : ', result[1])
print('부족수의 합 : ', result[2])
```

---

2. 자연수 N을 입력 받아, N개의 1 이상 3 이하의 난수로 이루어진 리스트 A를 생성한다. 리스트 A를 매개변수로 전달받는 함수 sumFour(a)를 사용하여 프로그램을 작성하라. sumFour(a)의 수행 과정은 다음과 같다.

- (1) 리스트 a의 원소들을 4개씩 묶은 다음, 4개의 원소 중에서 첫 번째 원소와 세 번째 원소의 합 s1과 두 번째 원소와 네 번째 원소의 합 s2를 구한다.
- (2) 새로운 리스트 b에 s1과 s2를 각각 순서대로 삽입한다.
- (3) 리스트 a의 모든 원소에 대한 처리가 끝나면 리스트 b를 반환한다.

- 순서도는 sumFour(a)에 대해 그린다.

-----

```
>>>
```

```
N = 8
```

```
A = [2, 2, 2, 1, 2, 2, 3, 3]
```

```
B = [4, 3, 5, 5]
```

```
>>>
```

```
N = 9
```

```
A = [2, 1, 2, 3, 3, 1, 3, 1, 1]
```

```
B = [4, 4, 6, 2, 1]
```

```
>>>
```

```
N = 10
```

```
A = [1, 3, 1, 3, 1, 2, 1, 1, 2, 3]
```

```
B = [2, 6, 2, 3, 2, 3]
```

```
>>>
```

-----

-----

```
def sumFour(a):
```

```
    b = []
```

```
    i = 0
```

```
    while i < N:
```

```
        s1 = a[i] + a[i+2]
```

```
        s2 = a[i+1] + a[i+3]
```

```
        b.append(s1)
```

```
        if s2 != 0:
```

```
            b.append(s2)
```

```
        i += 4
```

```
    return b
```

```
import random
N = int(input('N = '))
A = []
for i in range(N):
    A.append(random.randint(1, 3))
print('A =', A)
n = N
while n % 4 != 0:
    A.append(0)
    n += 1
print('B =', sumFour(A))
```

---

3. 숫자와 문자가 혼합되어 있는 문자열에서 연속으로 나오는 숫자를 모아서 리스트 a에 삽입한 다음, 리스트 a를 반환하는 함수 makeNumbers(s)를 사용하여 프로그램을 작성하라.

- 순서도는 makeNumbers(s)에 대해 그린다.
- 숫자와 문자 외에 특수문자가 있는 경우 '숫자와 문자만 입력 가능합니다.'라는 오류 메시지를 출력한다.

```
>>>
S = 34&654
숫자와 문자만 입력 가능합니다.
S = 123$32!34()
숫자와 문자만 입력 가능합니다.
S = 123adgt32df3de21
숫자 리스트 : ['123', '32', '3', '21']
>>>
```

숫자와 문자 외에 특수문자가 있는 경우 False를 반환하는 함수 isAlnum(s)와 숫자이면 True를 반환하는 함수 isNumber(s)는 다음과 같다.

```
def isAlnum(s):
    for i in range(len(s)):
        if not (s[i].isalnum()):
            return False
    return True

def isNumber(s):
    if ord(s) >= 48 and ord(s) <= 57:
        return True
    else:
        return False
```

```
def isAlnum(s):
    for i in range(len(s)):
        if not (s[i].isalnum()):
```

```
        return False
    return True
```

```
def isNumber(s):
    if ord(s) >= 48 and ord(s) <= 57:
        return True
    else:
        return False
```

```
def makeNumbers(s):
    i = 0
    n = len(s)
    b = []
    while i < n:
        a = ''
        while i < n and isNumber(s[i]):
            a += s[i]
            i += 1
        b.append(a)
        while i < n and not isNumber(s[i]):
            i += 1
    return b
```

```
S = input('S = ')
while not isAlnum(S):
    print('숫자와 문자만 입력 가능합니다.')
    S = input('S = ')
B = makeNumbers(S)
print('숫자 리스트 :', B)
```

---

4. 원소의 개수 N과 최소값 MIN, 최대값 MAX를 입력 받은 다음, N개의 MIN 이상 MAX 이하인 난수를 생성하여 리스트 A에 추가한다.

D = MAX - MIN 이라고 할 때, 리스트 A와 D를 매개변수로 전달받는 함수 countMinMax(a, d)를 사용하여 프로그램을 작성하라. countMinMax(a, d)는 리스트 a에 있는 난수의 빈도수를 세어 리스트 b에 추가한 다음 리스트 b를 반환한다.

- 순서도는 countMinMax(a, d)에 대해 그린다.

- 반환 받은 리스트 b = [24, 27, 26, 29, 21, 23] 라고 할 때, 다음과 같이 출력한다.

>>>

N = 150

MIN = 5

MAX = 10

난수	빈도수	비율
5	24	16.0%
6	27	18.0%
7	26	17.3%
8	29	19.3%
9	21	14.0%
10	23	15.3%

>>>

```
def countMinMax(a, d):  
    b = []  
    for i in range(d+1):  
        b.append(0)  
    for i in range(N):  
        b[a[i]-d] += 1  
    return b
```

```
import random  
N = int(input('N = '))  
MIN = int(input('MIN = '))  
MAX = int(input('MAX = '))
```



```
D = MAX - MIN
A = []
for i in range(N):
    A.append(random.randint(MIN, MAX))
B = countMinMax(A, D)
print()
print('난수\빈도수\비율')
print('=====')
for i in range(D+1):
    print('%d\t%d\t%.1f%%'%(i+D, B[i], B[i]*100/N))
```

---