

1. 두 개의 정수 A, B를 입력 받은 다음 두 정수를 더해서 변수 C에 저장한다. 이 변수에 저장된 정수를 한 자리씩 문자로 변환하여 리스트에 삽입한 다음, 이를 반환하는 함수 `makeString(c)`를 사용하여 프로그램을 작성하라.

- 이 문제에서 순서도는 `makeString(c)`에 대해서만 그린다.
- 정수를 문자열로 변환할 때 `str()` 함수를 사용하지 않는다.

>>>

A = 54632

B = 123456

A + B = 178088

문자열 리스트 : ['1', '7', '8', '0', '8', '8']

>>>

2. 다음과 같이 0과 1로 이루어진 정방행렬에서 i 를 행, j 를 열이라고 할 때, 사용자로부터 정방행렬의 크기 N 을 입력 받아 0과 1을 입력한다. 행렬을 나타내는 이차원 리스트를 A 라고 할 때, 다음과 같은 동작을 수행하는 함수 `exchangeMatrix(a)`를 사용하여 프로그램을 작성하라.

- (1) 상위 삼각 행렬($i < j$ 인 원소)에 있는 원소와 하위 삼각 행렬($i > j$)에 있는 원소가 대칭($a[i][j] = a[j][i]$)인 원소의 개수를 세어 변수 $c1$ 에 저장하고, 대칭이 아닌($a[i][j] \neq a[j][i]$) 원소의 개수를 세어 변수 $c2$ 에 저장한다.
- (2) 대칭이 아닌 원소는 서로 교환한다.
- (3) 모든 처리를 마치면 $c1$, $c2$, a 를 반환한다.

이 프로그램을 작성하는데 다음과 같은 2개의 함수를 사용한다.

- (1) 함수 `makeMatrix(n)` : $n \times n$ 행렬을 2차원 리스트 a 로 생성한 다음, 리스트 a 를 반환한다. 주대각선에는 0을 입력하고, 나머지 원소에는 0과 1을 입력한다.
- (2) 함수 `printMatrix(a)` : 2차원 리스트에 있는 원소를 행렬 모양으로 출력한다.

- 이 문제에서 순서도는 `exchangeMatrix(a)`에 대해서만 그린다.
- 0과 1은 난수를 발생시켜서 입력하고, 주대각선($i = j$)에 있는 원소에는 0을 입력한다.

>>>

N = 4

```
0 1 0 1
1 0 1 0
0 0 0 1
1 1 1 0
```

대칭인 원소의 개수 : 4

대칭이 아닌 원소의 개수 : 2

```
0 1 0 1
1 0 0 1
0 1 0 1
1 0 1 0
>>>
```

3. 다음과 같은 파일에서 "mailto:" 문자열 뒤에 있는 이메일 주소를 가져와서 리스트 a에 추가하는 함수 emailSearch(t, p)를 사용하여 프로그램을 작성한다. 모든 처리를 마치면 emailSearch(t, p)는 리스트 a를 반환한다. 이메일 주소는 "mailto:" 이후부터 쌍따옴표(")가 나올 때까지의 문자열이라고 생각하라. 문자열 탐색을 할 때는 함수 stringSearch()를 사용한다.

- 이 문제에서 순서도는 emailSearch(t, p)에 대해서만 그린다.
- 이메일 주소가 저장되어 있는 파일의 내용은 다음과 같다.

[email.txt]

```
<html>
<body>
  <ul>
    <li> <a href="mailto:gdhong@hanmail.net">Gildong Hong</a>
    <li> <a href="mailto:gsjang@gmail.com">Gilsan Jang</a>
    <li> <a href="mailto:yhkim@naver.com">Younghee Kim</a>
  </ul>
</body>
</html>
```

- 이 프로그램을 실행하면 다음과 같이 이메일이 출력된다.

```
gdhong@hanmail.net
gsjang@gmail.com
yhkim@naver.com
```

- 문자열 탐색을 수행하는 함수 stringSearch(t, p, i)는 다음과 같다.

```
def stringSearch(t, p, i):
    j = 0
    N = len(t)
    M = len(p)
    while i < N and j < M:
```

```
    if t[i] != p[j]:  
        i = i - j  
        j = -1  
    i += 1  
    j += 1  
if j == M:  
    return i - M  
else:  
    return i
```

4. 숫자와 '+', '-' 연산자가 혼합되어 있는 문자열에 대해 다음과 같은 동작을 수행하는 함수 makeNumbers(s)를 사용하여 프로그램을 작성하라.

(1) 연속으로 나오는 숫자를 모아서 정수로 만든다. 여기서 int() 함수를 사용하지 않는다.

(2) 정수를 리스트 a에 삽입한다.

(3) '+', '-' 연산자는 그대로 리스트 a에 삽입한다.

(3) 모든 처리를 마치면 리스트 a를 반환한다.

- 순서도는 makeNumbers(s)에 대해 그린다.

- 숫자와 '+', '-' 연산자 외의 문자가 있는 경우 '숫자와 +, - 기호만 입력 가능합니다.'라는 오류 메시지를 출력한다.

S = abc+234

숫자와 +, - 기호만 입력 가능합니다.

S = 234+76-34+45

수식 리스트 : [234, '+', 76, '-', 34, '+', 45]

계산 결과 : 321

숫자와 '+', '-' 연산자 외의 문자가 있는 경우 False를 반환하는 함수 isExpression(s), 숫자이면 True를 반환하는 함수 isNumber(s), 리스트에 있는 숫자와 '+', '-' 연산자를 사용하여 수식을 계산하는 함수 evalList(a)는 다음과 같다.

```
def isExpression(s):
```

```
    for i in range(len(s)):
```

```
        if not (isNumber(s[i]) or s[i] == '+' or s[i] == '-):
```

```
            return False
```

```
    return True
```

```
def isNumber(s):
```

```
    if ord(s) >= 48 and ord(s) <= 57:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def evalList(a):
```

```
    while len(a) > 1:
```

```
if a[1] == '+':  
    res = a[0] + a[2]  
elif a[1] == '-':  
    res = a[0] - a[2]  
a.pop(0)  
a.pop(0)  
a.pop(0)  
a.insert(0, res)  
return a[0]
```
