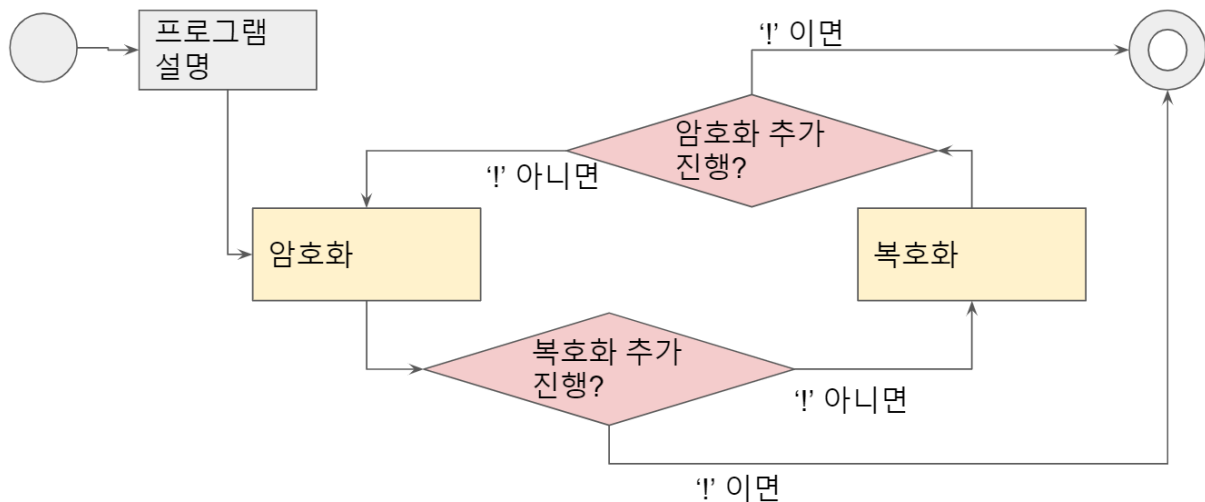


## 기말고사- 암호화/복호화

Comment는 점수에 포함되지는 않으나, 부분점수 채점을 위해 #문제 몇 번의 무엇 이렇게 간단히 적어주세요.

사용자가 '!' (느낌표)를 입력할 때까지 파일을 읽어 암호화/복호화 하는 것을 계속 반복하는 프로그램을 작성하라. 절차는 다음과 같다. 문제를 주의 깊게 읽고 절차를 준수하라. 아래의 절차대로 진행되지 않거나, 불필요한 절차가 추가된 경우 감점이다.



1. 프로그램을 처음 실행하면 이 프로그램 사용법을 200자 이상 안내하라. (내용에 점수 있음)
2. 처음은 무조건 암호화 모드이다.
  - A. 암호화할 txt 파일 이름을 확장자를 포함하여 입력 받고, 파일을 읽는다. 확장자 검사 는 하지 않는다.
    - i. 파일 읽기가 실패하면, 에러메시지를 출력한 후, 파일 이름을 다시 입력 받는다. 파일 읽기 성공할 때까지 계속 반복한다.
    - ii. 파일 읽기가 성공하면, 성공했다는 메시지와 함께 읽은 파일의 크기 (글자 수)를 출력한다. 다음 순서로 넘어간다.
  - B. ☒ 파일 내용으로부터 고유 글자(unique characters)를 추출한 후, 화면에 출력하라. 첫째 줄에는 출력할 내용에 대한 설명을 출력하고, 둘째 줄에 고유 글자들을 출력한다.
  - C. 고유 글자들을 랜덤하게 섞은 후, 원래 글자들과 순서대로 짝을 지어 암호화

dictionary를 만들고, 화면에 출력하라. 첫째 줄에는 출력할 내용에 대한 설명을 출력하고, 둘째 줄에 딕셔너리를 출력한다.

(예. 파일로부터 추출한 고유 글자들이 'a', 'b', 'c', 'd' 라면, 이 글자들을 랜덤하게 섞은 리스트 - 예를 들어 'b', 'd', 'a', 'c' - 를 만들고, 이것을 순서대로 짝을 지은 {'a': 'b', 'b': 'd', 'c': 'a', 'd': 'c'} 이 암호화를 위한 dictionary가 된다.)

D. 암호화 딕셔너리를 이용하여 파일 내용을 암호화한 후 '.enc' 파일로 저장하라. 파일 이름은 txt 파일이름에서 확장자를 뺀 나머지 부분을 이용한다. 파일 저장까지 끝난 후 암호화가 성공적으로 끝났다는 메시지와 함께 암호화된 내용의 전체 크기 (enc 파일의 글자 수)를 출력한다. enc 파일의 글자 수는 txt 파일의 글자 수와 동일해야한다.

E. 암호화 딕셔너리를 key와 value를 순서대로 파일에 저장한다. (예를 들어 딕셔너리가 {'a': 'b', 'b': 'd', 'c': 'a', 'd': 'c'} 라면 파일에는 abbdccadc 가 저장됨) 파일의 확장자는 '.key' 이고, 파일 이름은 txt 파일이름에서 확장자를 뺀 나머지 부분을 이용한다.

3. 암호화가 끝나면 복호화를 진행할 것인지 사용자에게 물어본다. 사용자가 '!'를 입력하면 프로그램을 종료하고, 그렇지 않으면 복호화 진행한다.

4. 복호화는 다음과 같이 한다.

A. 복호화할 enc 파일 이름을 확장자를 포함하여 입력 받고 (확장자 검사는 하지 않는다), enc 파일과 더불어 동일한 이름에 확장자만 .key를 가지는 암호코드 파일을 같이 읽는다.

i. 둘 중 하나라도 파일 읽기가 실패하면, 에러메시지를 출력한 후 파일 이름을 다시 입력받는다. 파일 읽기 성공할 때까지 계속 반복한다.

ii. 두 파일 모두 읽기 성공하면, 성공했다는 메시지와 함께 읽은 enc 파일의 크기 (글자 수)를 출력한다. 다음 순서로 넘어간다.

B. key 파일로부터 복호화 딕셔너리를 복구하여 화면에 출력한다. 복호화 딕셔너리는 암호화 딕셔너리와 아이템의 순서는 같되, key와 value 가 반대로 된 것이다. 첫째 줄에는 출력할 내용에 대한 설명을 출력하고, 둘째 줄에 딕셔너리를 출력한다.

C. 복호화 딕셔너리를 이용하여 파일 내용을 복호화한 후 '.dec' 파일로 저장하라. 파일 이름은 enc 파일이름에서 확장자를 뺀 나머지 부분을 이용한다.

i. 파일 저장까지 끝난 후 복호화가 성공적으로 끝났다는 메시지와 함께 복호화된 내용의 전체 크기 (dec 파일의 글자 수)를 출력한다. dec 파일의 글자 수는 enc 파일의 글자 수와 동일해야한다.

- ii. 복호화 도중에 에러가 발생하면 복호화 실패 메시지를 출력한다. 이 때는 dec 파일이 생성되지 않는다.
- 5. 복호화가 성공이든 실패든 일단 끝나면, 암호화를 진행할 것인지 사용자에게 물어본다. 사용자가 '!'를 입력하면 프로그램을 종료하고, 그렇지 않으면 다시 암호화부터 진행한다.

<예시 화면 - 출력 화면이 너무 길어 뒷부분 생략>

```
Encryption/Decryption
file name to encrypt: alice.t
Error: I/O Error [Errno 2] No such file or directory: 'alice.t'
file name to encrypt: alice.txt
File Read Success. The size is 145190
The unique character set from alice.txt
{'J', '!', 'f', 'R', '"', '?', 'Y', '2', 'X', '(', 'w', '-', 'c', 'o', 'h', 'p',
 'b', ')', '\n', '*', ';', 'O', 'G', ' ', '3', ':', 'K', 'i', 'N', 'j', 'H', 'x',
 'T', 't', 'F', 'Z', 'a', ']', '_', 'z', 'D', 'C', 'B', '[', 'S', '9', 'e', 'y',
 'm', 'g', ' ', 's', 'k', 'Q', '8', 'l', '4', 'M', 'n', '0', 'E', 'u', 'W', 'r',
 'L', 'U', 'I', 'P', '5', 'A', '"', 'q', 'd', '.', '#', 'v', '1', '\uffeff', 'V'
}
Encryption Dictionary:
{'J': '"', 'I': 'M', 'f': ' ', 'R': 'i', '"': 'd', '?': 't', 'Y': '\n', '2': '2',
 'X': 'W', '(': ']', 'w': 'N', '-': 'f', 'c': 'w', 'o': ' ', 'h': 'o', 'p': '*'
 'b': '!', ')': 'Z', '\n': '(', '*': 'D', ' ': 'm', 'O': '2', 'G': 'k', ' ': 'P',
 '3': 'A', ':': '3', 'K': 'B', 'i': ':', 'N': 'u', 'j': 'J', 'H': ')', 'x': 'p',
 'T': '5', 't': 'C', 'F': 'V', 'Z': '4', 'a': 'z', ']: 'v', '_': 'a', 'z': '-',
 'D': '#', 'C': 's', 'B': 'I', '[': 'n', 'S': 'X', '9': ' ', 'e': 'Y', 'y': 'e',
 'm': 'R', 'g': ' ', ' ': 'x', 's': 'K', 'k': 'q', 'Q': 'T', '8': 'l', 'l': '\uffeff',
 '4': 'g', 'M': 'F', 'n': 'c', '0': '9', 'E': '"', 'u': 'O', 'W': 'h', 'r',
 ': 'U', 'L': 'H', 'U': 'j', 'I': 'L', 'P': 'E', '5': '[', 'A': 'l', '"': 'S', 'q',
 ': 'Q', 'd': 'y', '.': 'G', '#': 'r', 'v': '.', '1': '8', '\uffeff': '0', 'V': 'b'
}
Encryption succeeded. The size is 145190
Do you like to decrypt a file? If no, type !: s
file name to decrypt: alice.enc
File Read Success. The size is 145190
Decryption Dictionary:
{' ': 'J', 'M': '!', ' ': 'f', 'i': 'R', 'd': '"', 't': '?', '\n': 'Y', '2': '2',
 'W': 'X', ']: '(', 'N': 'w', 'f': '-', 'w': 'c', ' ': 'o', 'o': 'h', '*': 'p',
 'I': 'b', 'Z': ')', '(': '\n', 'D': '*', 'm': ';', '2': 'O', 'k': 'G', 'P': ' ',
 'A': '3', '3': ':', 'B': 'K', ':': 'i', 'u': 'N', 'J': 'j', ')': 'H', 'p': 'x',
 '5': 'T', 'C': 't', 'V': 'F', '4': 'Z', 'z': 'a', 'v': ']', 'a': ' ', '-': 'z',
 '#': 'D', 's': 'C', 'I': 'B', 'n': '[', 'X': 'S', ';': '9', 'Y': 'e', 'e': 'y',
 'R': 'm', ' ': 'g', 'x': ' ', 'K': 's', 'q': 'k', 'T': 'Q', 'l': '8', '\uffeff',
 ': 'l', 'g': '4', 'F': 'M', 'c': 'n', '9': '0', '"': 'E', 'O': 'u', 'h': 'W', 'U',
 ': 'r', 'H': 'L', 'j': 'U', 'L': 'I', 'E': 'P', '[': '5', 'l': 'A', 'S': '"', 'Q',
 ': 'q', 'y': 'd', 'G': '.', 'r': '#', '.': 'v', '8': '1', '0': '\uffeff', 'b': 'V'
}
Decryption Succeeded. The size is 145190
Do you like to encrypt a file? If no, type !: k
file name to encrypt: The_little_prince.txt
File Read Success. The size is 90565
The unique character set from The_little_prince.txt
{'$', 'J', '!', 'f', 'R', '"', '?', 'Y', '2', '(', 'w', '-', 'c', 'h', 'o', 'p',
 'b', ')', '\n', ';', 'O', ' ', 'G', '>', '3', ':', '6', 'K', 'i', 'N', '7', 'j',
 'x', 'H', 'T', 't', 'F', 'Z', 'a', '<', 'z', 'D', 'C', 'B', 'S', '9', 'e', 'y',
 'm', 'g', ' ', 's', 'k', '8', 'l', '\u3000', '4', 'M', 'n', '0', 'E', 'u', 'W',
 'r', 'L', 'U', 'I', 'P', '5', 'A', '"', 'q', 'd', '.', 'v', '1', 'V'}
Encryption Dictionary:
```