

# 소프트웨어프로젝트 2주차 주간보고서

컴퓨터SW 18017103 황제현

# 목차

- 1. 이번주차 진행내용
  - 2. 진행상황
- 3. 첨부) 파이썬 코드 파일

# 1. 이번주차 진행내용

- 암호화 프로그램 중 평문 암호화 단계의 초기 버전 작성

## 2. 진행상황

- 일부 변수명을 수정하고,  
필요한 데이터를 추가 정의하였다.

```
plainText = "" # 암호화할 데이터
keyString = "" # 암호,복호화에 사용할 키
keyList = [] # 연산을 위해 리스트화시킨 키
ConvertedKeyList = [] # 진법변환된 키
conversionNum = 0 # 변환할 진법
InputEncodeType = sys.stdin.encoding
OutputEncodeType = sys.stdout.encoding

UniConvertedData = [] # 유니코드로 변환된 데이터
NumConvertedData = [] # 진법 변환된 데이터
cipherText = "" # 암호화된 데이터
cipherList = []
```

## 2. 진행상황

- 데이터를 입력받고 유니코드로 변환하는 단계
- 실행결과를 확인하기 위해 print문을 추가하였다.

```
# =====Test Code=====
# 원본 데이터 확인
print("원본 데이터 : ", end="")
for i in range(0, len(plainText)) :
    print(plainText[i], end=" ")
print()

# 유니코드로 변환하여 데이터 저장
for i in range(0, len(plainText)) :
    UniConvertedData.append(ord(plainText[i]))

# 변환된 데이터 확인
print("유니코드로 변환 : ", end="")
for i in range(0, len(UniConvertedData)) :
    print(UniConvertedData[i], end=" ")
print()
```

소스코드 일부

```
변환할 데이터를 입력하세요: suction 2735
원본 데이터 : suction 2735
유니코드로 변환 : 115 117 99 116 105 111 110 32 50 55 51 53 |
```

실행결과

## 2. 진행상황

- 유니코드로 변환된 평문 데이터를  
진법 변환하는 단계
- 용이한 연산을 위해 리스트에 저장  
하여 다루었다.

```
# 진법변환 프로세스 1
print("=====진법변환 프로세스(유니코드 데이터)=====")
conversionNum = random.randrange(5,10) # 5~9 사이의 진법 중 랜덤으로 결정.
print("선택된 진법: ", conversionNum)

dataLength = len(UniConvertedData) # 변환할 데이터 길이

for i in range(0, len(UniConvertedData)) :
    tmpList = [] # 진법변환된 숫자 하나를 저장할 리스트
    tmpStr = ""
    tmpNum = UniConvertedData[i]
    while True :
        tmpList.append(tmpNum % conversionNum)
        tmpNum //= conversionNum
        if(tmpNum == 0) : break
    tmpList.reverse()
    for i in tmpList :
        tmpStr += str(i)
    NumConvertedData.append(int(tmpStr))

print("진법변환된 유니코드 데이터 : ", end="")
print(NumConvertedData)
```

소스코드 일부

```
=====진법변환 프로세스(유니코드 데이터)=====
선택된 진법: 5
진법변환된 유니코드 데이터 : [430, 432, 344, 431, 410, 421, 420, 112, 200, 210, 201, 203]
```

실행결과

## 2. 진행상황

- random.randrange()를 사용하여 무작위로 키를 생성하고, 같은 진법으로 변환하였다.

```
=====키 생성=====
[96, 28, 54, 45, 18, 25, 67, 69, 35, 4, 51, 64]
```

```
=====진법변환 프로세스(키 데이터)=====
진법변환된 키 데이터 : [341, 103, 204, 140, 33, 100, 232, 234, 120, 4, 201, 224]
```

```
print()
print("=====키 생성=====")
for i in range(0, len(UniConvertedData)) :
    keyList.append(random.randrange(0,99))

print(keyList)
print()

# 진법변환 프로세스 2
print("=====진법변환 프로세스(키 데이터)=====")
for i in range(0, len(keyList)) :
    tmpList = [] # 진법변환된 숫자 하나를 저장할 리스트
    tmpStr = ""
    tmpNum = keyList[i]
    while True :
        tmpList.append(tmpNum % conversionNum)
        tmpNum //= conversionNum
        if(tmpNum == 0) : break
    tmpList.reverse()
    for i in tmpList :
        tmpStr += str(i)
    ConvertedKeyList.append(int(tmpStr))

print("진법변환된 키 데이터 : ", end="")
print(ConvertedKeyList)
print()
```

## 2. 진행상황

- 변환된 평문과 키를 XOR연산하고 문자로 출력하는 단계
- 숫자가 커서 특수문자로만 변환되는 문제점이 있었다.
- 다양한 문자로 변환될 수 있도록 연산방식을 조금 개선할 계획이다.

```
=====암호화 프로세스=====
XOR연산된 리스트 : [251, 471, 404, 291, 443, 449, 332, 154, 176, 214, 0, 43]
û, Û, Ÿ, ġ, ž, ll, Ō, , °, Ŏ, , +,
kString: ûÛŸġžllŌ°Ŏ +
tl : [251, 471, 404, 291, 443, 449, 332, 154, 176, 214, 0, 43]
cipherList [251, 471, 404, 291, 443, 449, 332, 154, 176, 214, 0, 43]
```

```
print("=====암호화 프로세스=====")
# 유니코드 데이터와 키 데이터를 진법변환해서 XOR연산.
for i in range(0, len(UniConvertedData)) :
    cipherList.append(NumConvertedData[i] ^ ConvertedKeyList[i])
print("XOR연산된 리스트 : ", cipherList)

kString = ""
for i in range(0, len(UniConvertedData)) :
    print(chr(cipherList[i]), end=', ')
    kString += chr(cipherList[i])

print()
print("kString: ", kString)

tl = []
for i in range(0, len(kString)) :
    tl.append(ord(kString[i]))

print("tl : ", tl)
print("cipherList ", cipherList)
```

소스코드

실행결과