

1. 문자열을 입력 받아 반대로 출력하는 프로그램을 작성하자.

```
## 변수 선언 부분##
inStr, outStr = "", ""
count, i = 0, 0

## 메인 코드 부분##
inStr = input("문자열을 입력하세요: ")
count = len(inStr)

for i in range(0, count) :

    print("내용을 거꾸로 출력--> %s" % outStr)
```

2. 입력된 문자열 중에서 숫자를 제거하는 프로그램을 작성하자.

예) 문자열 --> Sunrin Internet 1234 파일썬

숫자 제거 --> Sunrint Internet 파일썬

```
inStr, outStr = '', ''  
inStr = input('문자열 -->')  
for i in range(0, len(inStr)) :  
  
    print('숫자 제거 --> ' + outStr)
```

3. 문자열이 괄호로 감싸 있지 않으면 괄호로 감싸 주는 프로그램을 작성해보자.

```
ss = input("입력 문자열==> ")
print("출력 문자열==> ", end = '')

if ss._____ == False :
    print("(", end = '')

print(ss, end = '')

if ss._____ == False :
    print(")", end = '')
```

```

inStr = "    한글Python 프로그래밍"
outStr = ""

for i in range(0, len(inStr)) :
    if inStr[i] != ' ' :

    _____


print("원래 문자열=> " + '[' + inStr + ']')
print("공백 삭제 문자열=> " + '[' + outStr + ']')

```

4. 연/월/일 형식으로 문자열을 입력받아 10년 후 날짜를 출력하는 코드를 작성해보자.

```

ss = input("날짜(연/월/일) 입력=> ")

ssList = ss._____

print("입력한 날짜의 10년 후=> ", end = '')
print(str(int(ssList[0]) + 10) + "년", end = '')
print(ssList[1] + "월", end = '')
print(ssList[2] + "일")

```

5. 입력한 문자열에서 숫자, 영문 소문자, 영문 대문자, 한글, 기타 문자의 개수를 세는 프로그램을 작성하자.

힌트) 각 문자는 고유한 번호가 할당되어 있으며, `ord()` 함수로 번호를 확인할 수 있다.

[출력 결과]

문자열을 입력하세요 : Sunrin Internet 소프트웨어과 2학년입니다. ^__^

대문자 : 2 소문자: 12 숫자: 1 한글: 11 기타: 10

```

## 전역 변수 선언 부분
numCnt, lowerCnt, upperCnt, hanCnt, etcCnt = [0] * 5
ch = ""

## 메인(main) 코드 부분
inStr = input("문자열을 입력하세요 : ")
for ch in inStr :
    if ( ord(ch) >= ord("A") and ord(ch) <= ord("Z")) :
        upperCnt += 1
    elif _____

```

```

lowerCnt += 1
elif _____
    numCnt += 1
elif _____
    hanCnt += 1
else :
    etcCnt += 1
print("대문자 :", upperCnt, " 소문자:", lowerCnt, " 숫자:", numCnt, " 한글:", hanCnt, " 기타:",
etcCnt)

```

6. 다음 code-1을 map 함수를 활용하여 해결하시오.

code-1

```

def add_1(n):
    return n + 1

target = [1, 2, 3, 4, 5]
result = []
for value in target:
    result.append(add_1(value))

print(result) # 출력결과 : [2, 3, 4, 5, 6]

```

code-2

```

def add_1(n):
    return n + 1

target = [1, 2, 3, 4, 5]

result = _____

print(result)

```

7. <보기>의 코드를 다음 조건에 만족하도록 작성하시오.

- (1) 숫자1, 연산자, 숫자2 순서로 입력받는다.
- (2) 제곱(**) 연산자를 추가한다.
- (3) 0으로 나누려고 하면 메시지를 출력하고 계산되지 않도록 한다.

<보기>

```
## 함수 선언 부분##
def calc(v1, v2, op) :
    result = 0
    if op == '+' :
        result = v1 + v2
    elif op == '-' :
        result = v1 - v2
    elif op == '*' :
        result = v1 * v2
    elif op == '/' :
        result = v1 / v2
    return result

## 전역 변수 선언 부분##
res = 0
var1, var2, oper = 0, 0, ""

## 메인 코드 부분##
oper = input("계산을 입력하세요(+, -, *, /) : ")
var1 = int(input("첫 번째 수를 입력하세요: "))
var2 = int(input("두 번째 수를 입력하세요: "))

res = calc(var1, var2, oper)

print("## 계산기: %d %s %d = %d" % (var1, oper, var2, res))
```

8. 다음 코드의 실행결과를 작성하시오.

```
## 함수 선언 부분##
def func1() :
    global a      # 이 함수 안에서a는 전역 변수
    a = 10
    print("func1()에서a값%d" % a)

def func2() :
```

```
print("func2()에서 a값%d" % a)
```

```
## 함수 변수 선언 부분##  
a = 20          # 전역 변수
```

```
## 메인 코드 부분##  
func1()  
func2()
```

실행결과

9.

```
## 함수 선언 부분##  
def func1() :  
    result = 100  
    return result  
  
def func2() :  
    print("반환값이 없는 함수 실행")  
  
## 전역 변수 선언 부분##  
hap = 0  
  
## 메인 코드 부분##  
hap = func1()  
print("func1()에서 돌려준 값==> %d" % hap)  
func2()
```

실행결과

10. 다음 코드를 실행한 결과를 작성하시오.

```
## 함수 선언 부분##  
def multi(v1, v2) :  
    retList=[]      # 반환할 리스트
```

```

res1 = v1 + v2
res2 = v1 - v2
retList.append(res1)
retList.append(res2)
return retList

## 전역 변수 선언 부분##
myList = []
hap, sub = 0, 0

## 메인 코드 부분##
myList = multi(100, 200)
hap = myList[0]
sub = myList[1]
print("multi()에서 돌려준 값==> %d, %d" % (hap, sub))

```

실행결과

11. 매개변수의 개수를 지정하지 않는 함수를 작성하시오.

```

## 함수 선언 부분##
def para_func(_____) :
    result = 0
    for num in para :
        result = result + num

    return result

## 전역 변수 선언 부분##
hap = 0

## 메인 코드 부분##
hap = para_func(10, 20)
print("매개변수가2개인 함수를 호출한 결과==> %d" % hap)
hap = para_func(10, 20, 30,40,50,60,70,80,90,100)
print("매개변수가10개인 함수를 호출한 결과==> %d" % hap)

```

실행결과

매개변수가10개인 함수를 호출한 결과==> 5050

12. 딕셔너리 형식의 매개변수를 처리하는 함수를 작성해보자.

```
def dic_func(______):
    for k in para.keys():
        print("%s -> %d명입니다." %(k, para[k]))

grade={'1st':178, '2nd':200, '3rd':198}
dic_func(______)
```

실행결과

1st -> 178명입니다.

2nd -> 200명입니다.

3rd -> 198명입니다.