**[ 실습 1 ]**

**1. 파일명 : funcLab8.py**

**2. 구현해야 하는 함수 사양**

**함수명 : print\_triangle\_withdeco**

**매개변수 : 2개**

**숫자와 데코문자**

**여기에서 데코문자는 기본값을 갖는다. 기본값은 ‘%’로 정한다.**

**리턴값 : 없음**

**기능 : 전달된 숫자 개수로 구성되는 삼각형을 출력한다. 출력 형식은 다음과 같다.**

**숫자 2 만 전달시**

**%**

**%%**

**숫자 5 와 데코문자 ‘\*’ 전달시**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**전달되는 아규먼트 값은 1~10으로 제한한다. 1~10 이외의 값이 전달된 경우에는**

**처리하지 않는다.**

**3. 숫자를 다양하게 지정해서 print\_triangle\_withdeco () 함수를 호출해 본다.**

**[ 실습 2 ] - 최댓값을 구하는 기능은 함수를 사용하지 않고 제어문으로 직접 구현한다.**

**1. listLab1.py 이라는 소스를 생성한다.**

**2. 다음 값들로 구성되는 리스트를 생성하여 listnum 에 저장한다.**

**10, 5, 7, 21, 4, 8, 18**

**3. listnum 에 저장된 값들 중에서 최댓값을 추출하여 다음과 같이 출력한다.**

**최댓값 : 21**

**[ 실습 3 ] - 최솟값을 구하는 기능은 함수를 사용하지 않고 제어문으로 직접 구현한다.**

**1. listLab2.py 이라는 소스를 생성한다.**

**2. 다음 값들로 구성되는 리스트를 생성하여 listnum 에 저장한다.**

**10, 5, 7, 21, 4, 8, 18**

**3. listnum 에 저장된 값들 중에서 최솟값을 추출하여 다음과 같이 출력한다.**

**최솟값 : 4**

**[ 실습 4 ] - 최댓값과 최솟값을 구하는 기능은 함수를 사용하지 않고 제어문으로 직접 구현한다.**

**1. listLab3.py 이라는 소스를 생성한다.**

**2. 다음 값들로 구성되는 리스트를 생성하여 listnum 에 저장한다.**

**10, 5, 7, 21, 4, 8, 18**

**3. listnum 에 저장된 값들 중에서 최댓값 최솟값을 추출하여 다음과 같이 출력한다.**

**최솟값 : 4, 최댓값 : 21**

**[ 실습 5 ]**

**1. listLab4.py 이라는 소스를 생성한다.**

**2. 비어있는 리스트를 하나 생성하여 listnum 이라는 변수에 저장한다.**

**3. 1~50 사이의 난수를 10개 추출하여 listnum 에 추출 순서대로 저장한다. (for문 사용)**

**4. listnum의 모든 값들을 출력한다.(이 때 반복문을 사용하지 않아도 된다.)**

**5. 리스트에서 10보다 작은 값들은 100으로 변경한다. (for문 사용)**

**6. listnum의 모든 값들을 출력한다.(이 때 반복문을 사용하지 않아도 된다.)**

**7. 인덱싱 방법으로 listnum의 첫 번째 데이터를 출력한다.**

**8. 인덱싱 방법으로 listnum의 마지막 데이터를 출력한다.**

**9. 슬라이싱 방법으로 listnum의 두 번째 데이터부터 여섯 번째 데이터만 추출하여**

**출력한다.**

**10. 슬라이싱 방법으로 listnum의 데이터를 역순으로 출력한다.**

**11. 슬라이싱 방법으로 listnum의 데이터를 모두 출력한다.**

**12. 인덱싱 방법으로 5번째 데이터를 삭제한다.**

**13. 슬라이싱 방법으로 listnum의 데이터를 모두 출력한다.**

**14. 슬라이싱 방법으로 2~3번째 데이터를 삭제한다.**

**15. 슬라이싱 방법으로 listnum의 데이터를 모두 출력한다.**

**[ 실습 6 ]**

**1. 파일명 : funcLab9.py**

**2. 구현해야 하는 함수 사양**

**함수명 : sumEven1**

**매개변수 : 가변형(전달받을 수 있는 아규먼트 개수에 제한이 없다.)**

**리턴값 : 1개**

**기능 : 아규먼트가 몇 개가 전달되든 처리해야 한다.**

**아규먼트는 1 이상의 숫자만 온다고 정한다.**

**전달된 아규먼트들에서 짝수에 해당하는 숫자들만 합을 계산해서 리턴한다.**

**전달된 아규먼트들 중에 짝수가 없으면 0을 리턴한다.**

**아규먼트가 전달되지 않으면 -1을 리턴한다.**

**3. 숫자를 다양하게 지정해서 sumEven1() 함수를 호출해 본다.**

**[ 실습 6 ]**

**1. 파일명 : funcLab10.py**

**2. 구현해야 하는 함수 사양**

**함수명 : sumAll**

**매개변수 : 가변형(전달받을 수 있는 아규먼트 개수에 제한이 없다.)**

**리턴값 : 1개**

**기능 : 아규먼트가 몇 개가 전달되든 처리해야 한다.**

**호출시 전달되는 아규먼트의 데이터 타입에는 제한이 없다. 그러므로 전달된 아규먼트의**

**타입을 채크하여 숫자만 처리하고 숫자가 아닌 데이터는 무시한다.**

**아규먼트가 전달되지 않았거나 전달되었다 하더라도 숫자가 없으면 None 을 리턴한다.**

**3. 숫자를 다양하게 지정해서 sumAll() 함수를 호출해 본다.**

**구현된 모든 소스들은 메일로 제출하며 첨부파일로 추가한다.**

**이 때 메일 제목은 “7월13일실습1-XXX” 로 작성하여 전송한다.**