2017년 1학기 국민대 과학과소프트웨어적사고 기말고사

일시: 2017년 6월 12일 19:30~21:00

|  |  |
| --- | --- |
| 점수: | 분반: 담당 교수: |
| 학번: 이름: |

모든 문제와 풀이는 파이선 3 버전의 문법을 사용합니다!

1. 다음 중 틀린 설명을 모두 고르시오. (20점)
2. 집합을 print로 출력하면 자료가 입력되었던 순서대로 출력된다. 정답
3. 함수 내부에서 리스트의 값을 변경하면 함수 종료 후에도 변경된 값이 유지된다.
4. 튜플은 내용 변경이 불가능하다.
5. if \_\_name\_\_ == "\_\_main\_\_": 를 넣어주지 않으면 모듈은 동작하지 않는다. 정답
6. 함수에서 global 표시가 없으면 전역 변수의 값을 읽지 못한다. 정답
7. 파일 객체의 readlines()는 파일 전체를 읽어서 하나의 스트링을 반환한다. 정답
8. 파일을 읽다가 맨 처음으로 이동하려면 seek(0) 함수를 사용한다.
9. 피클링을 사용하기 위해서는 파일을 반드시 이진모드로 열어야 한다.
10. 조화수열은 Zipf’s law를 따른다.
11. 예외 처리에서 finally는 예외가 발생하지 않은 경우 수행할 코드 블록이다. 정답
12. 밑줄 친 부분에 라무다 함수를 사용하여, 리스트가 튜플의 두 번째 값을 기준으로 정렬이 되도록 하시오 (10점).

|  |
| --- |
| a=[(“k2”, 2010), (“k3”, 2006), (“k1”, 2004)]  a.sort(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) 정답: key = lambda x:x[1] |

1. 다음은 사용자가 지정하는 파일을 읽고 파일에 등장하는 각 영어 단어의 개수를 세는 프로그램이다. 파일에는 순수한 영어 단어만이 공백문자로 구분되어 나열된다고 가정한다. 밑줄 친 곳을 채우시오 (15점).

|  |
| --- |
| fileName = input()  f = open(fileName, “r”)  myDict = {}  for line in f:  words = line. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 정답: split()  for w in words:  if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 정답: w not in myDict:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 정답: myDict[w] = 1  else:  myDict[w] += 1 |

1. “import math”와 “from math import \*”의 차이점을 기술하시오 (10점).
2. 지면에서 수직 위로 던진 물체의 t초 후의 속도는 아래 식을 따른다. 사용자로부터 정수 v0 값을 입력 받아서 물체가 땅에 떨어질 때까지 걸리는 시간을 계산하여 출력하는 간단한 프로그램을 작성하시오. 단, 사용자가 -1 을 입력할 때까지 프로그램은 계속 동작한다. 중력가속도 g는 9.8로 계산한다 (10점).
3. 다음은 파일의 내용을 읽어서 몬테카를로 방법을 적용한 코드이다. os.path.getsize()는 파일의 크기(바이트 수)를 리턴 한다. 파일 크기는 항상 짝수라고 가정한다. 밑줄 친 부분을 올바르게 수정하시오 (15점).

|  |
| --- |
| import math, os    f = open("normal.zip","rb")  n = os.path.getsize("normal.zip")    inside = 0  buf = f.read()  for i in range(1, 10000) : 정답: i in range(1, len(buf), 2)  x = buf[i-1]  정답: x = buf[i-1] / 255  y = buf[i]   정답: y = buf[i] / 255  if(math.sqrt(x \* x + y \* x) <= 1) :  inside += 1    f.close()    pi = 4 \* inside / n 정답: n 자리에 (n//2) 또는 (n/2)  print(pi) |

1. 다음 코드는 사용자로부터 디렉토리 이름을 입력 받아 해당 디렉토리와 모든 하위 디렉토리에 속하는 파일들의 사이즈를 모두 합산하는 프로그램이며 재귀함수로 작성되었다. 밑줄 친 부분을 채우시오. “os.path.getsize(fileName)” 함수는 fileName에 해당하는 파일의 사이즈를 리턴한다 (20점).

|  |
| --- |
| import os  def dirSize(n):  totalSize=0  if not os.path.isdir(n):  totalSize +=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 정답: totalSize +=os.path.getsize(n)  else:  fList = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 정답: os.listdir(n)  for s in fList:  totalSize += \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 정답: dirSize(n + ‘\’ + s)  return totalSize  name = input()  print(dirSize(name)) |