2019학년도 11월 실시 (정보처리와 관리) 문제해결실습 정답

2학년 (　　)반 번호 : (　　) 이름 : (　　)

※프로그램의 코드의 주석(comment)과 실행 결과 화면을 보고 코드의 빈 칸을 채우세요.

7차시_스파이럴그리기&데이터를묶어서표현하기

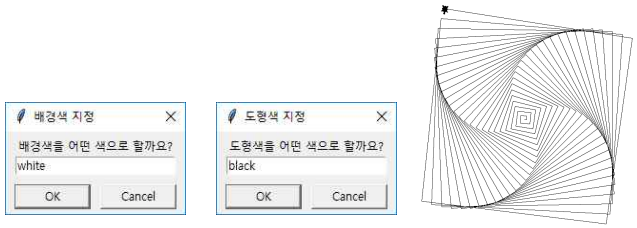
1. Ex1_선을반복해서스파이럴그리기(터틀)

[조건]
1.배경색과 도형색은사용자로부터 입력받는다.
2.선 그리기와 각도 조절을 100회 반복한다.
3.선의 초기값은 5픽셀이고 반복할 때마다 5씩 증가된다.
4.각도는 반복할 때마다 89도씩 왼쪽으로 회전한다.

코드

```
import turtle
t=turtle.Turtle()
t.shape("turtle")
turtle.title("스파이럴 그리기")
t.speed(0) #거북이의 속도를 가장 빠르게 지정

bcolor=turtle.textinput("배경색 지정","배경색을 어떤 색으로 할까요?") #배경색 입력
scolor=turtle.textinput("도형색 지정","도형색을 어떤 색으로 할까요?") #도형색 입력

turtle.bgcolor(bcolor) #배경색 지정하기
t.color(scolor) #도형색 지정하기

angle=89 #각도
line=5 #길이

for x in range(  (가)  ) : #100번 반복
    (나)  #line만큼 거북이가 앞으로 이동함
    t.left(angle) #왼쪽으로 angle만큼 회전함
    (다)  #반복하면서 line의 길이를 5씩 증가시킴

turtle.done()
```

실행 결과 화면



(가)	100
(나)	t.forward(line)
(다)	line=line+5

2019학년도 11월 실시 (정보처리와 관리) 문제해결실습 정답

2학년 (　　)반 번호 : (　　) 이름 : (　　)

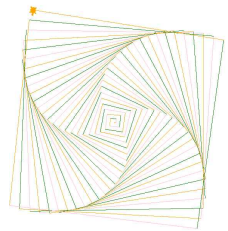
2. Ex2_선색을바꾸면서스파이럴그리기(터틀) : Ex1을 참고하여 [조건]을 추가하여 스파이럴을 변형하기

[조건]
0~99까지의 값을 활용하여 100번을 반복하면서
1. 만약 값이 3으로 나누어 나머지가 0과 같으면 도형의 색을 오렌지 색으로 바꾼다.
2. 만약 값이 3으로 나누어 나머지가 1과 같으면 도형의 색을 분홍색으로 바꾼다.
3. 만약 값이 3으로 나누어 나머지가 2과 같으면 도형의 색을 초록색으로 바꾼다.

코드

```
import turtle
t=turtle.Turtle()
t.shape("turtle")
turtle.title("스파이럴 그리기")
t.speed(0) #거북이의 속도를 가장 빠르게 지정

turtle.bgcolor("white") #배경색 지정하기

angle=89 #각도
line=5 #길이

for x in range(100) : #100번 반복 [0,1,2,3,...,97,98,99]
    if(  (가)  ) : #만약 x를 3으로 나눈 나머지가 0과 같으면
        t.color("orange") #도형의 색을 오렌지색으로 함
    if(  (나)  ) : #만약 x를 3으로 나눈 나머지가 1과 같으면
        t.color("pink") #도형의 색을 분홍색으로 함
    if(  (다)  : #만약 x를 3으로 나눈 나머지가 2와 같으면
        t.color("green") #도형의 색을 초록색으로 함

    t.forward(line) #line만큼 거북이가 앞으로 이동함
    t.left(angle) #왼쪽으로 angle만큼 회전함
    line=line+5 #반복하면서 line의 길이를 5씩 증가시킴

turtle.done()
```

실행 결과 화면



(가)	x%3==0
(나)	x%3==1
(다)	x%3==2

3. Ex3_내마음대로스파이럴
4. Ex4_무엇이든물어보살 : 사용자의 질문에 자동으로 대답해 주는 프로그램 작성

코드

```
import time    #time.sleep()을 사용하기 위한 모듈 가져오기
import [   (가)   ]  #random.choice()를 사용하기 위한 모듈 가져오기
wordList=["네, 확실합니다.",
          "전망이 좋은 것 같네요",
          "믿으셔도 됩니다",
          "글쎄요, 아닌 것 같군요",
          "한 점의 의심도 없습니다",
          "그럼요! 명백히 올바른 선택을 했습니다",
          "제 답변은 NO입니다"] #자동 대답을 위한 답변 리스트 작성

print("==무엇이든 물어보살==")
name=input("이름을 입력 : ")
question=input("무엇에 대해 알고 싶은가요?")

print(name,"님,",question,"에 대해 질문하셨군요.")
print("운명의 주사위를 굴려봅니다.")

for i in range([   (나)   ]): #[6,5,4,3,2,1]
    print(i)
    time.sleep(1) #1초씩 지연 시간을 갖게 한다.

print([   (다)   ](wordList)) #wordList에서 랜덤으로 하나의 값을 가져와 출력한다.

input()
```

실행 결과 화면

```
==무엇이든 물어보살==
이름을 입력 : 김경복
무엇에 대해 알고 싶은가요?대학에 잘 진학할 수 있을까요?
김경복 님, 대학에 잘 진학할 수 있을까요? 에 대해 질문하셨군요.
운명의 주사위를 굴려봅니다.
6
5
4
3
2
1
그럼요! 명백히 올바른 선택을 했습니다
```

(가)	random
(나)	6,0,-1
(다)	random.choice

2019학년도 11월 실시 (정보처리와 관리) 문제해결실습 정답
2학년 ( )반 번호 : ( ) 이름 : ( )

5. Ex5_타자게임

코드

```
import random
w=["cat","dog","fox","monkey","wolf",
      "mouse","panda","frog","snake"]#문제 리스트
n=1 #문제 번호
score=0 #점수
print("==타자게임==")
print("오타 없으면 +20점, 오타 발생 시 -5점")

while([   (가)   ]) : #문제는 5문제
    #문제 리스트(w)에서 랜덤으로 하나를 선택,q에 저장
    q=[   (나)   ]
    print("*문제",n)
    print(q) #문제 출력
    x=input() #사용자로부터 입력받기
    if([   (다)   ]) : #오타가 없을 경우
        score=score+20 #점수를 20점 증가한다.
        print("+20점!! 현재 스코어 : ", score)
    else : #오타가 있을 경우
        print("오타!!")
        score=score-5 #점수를 5점 감소한다.
        print("-5점!! 현재 스코어 : ",score)
    n=n+1 #문제 증가

print("최종 점수 :", score)

input()
```

```
==타자게임==
오타 없으면 +20점, 오타 발생 시 -5점
*문제 1
wolf
wolf
+20점!! 현재 스코어 : 20
*문제 2
snake
snke
오타!!
-5점!! 현재 스코어 : 15
*문제 3
fox
fox
+20점!! 현재 스코어 : 35
*문제 4
mouse
mouser
오타!!
-5점!! 현재 스코어 : 30
*문제 5
panda
panda
+20점!! 현재 스코어 : 50
최종 점수 : 50
```

(가)	n<=5
(나)	random.choice(w)
(다)	q==x

[8차시_랜덤으로수행하기]

1. Ex1_복권당첨확인하기

코드

```
import random
user=int(input("가지고 있는 복권 번호를 입력 : ")) #사용자 입력
lotto=random.[  (가)  ](10,99) #10~99사이의 숫자 중 랜덤으로 하나의 수 얻기
if([  (나)  ]) : #만약 사용자 입력값(user)과 랜덤으로 발생된 수(lotto)가 같으면
    print("당첨입니다!!")
else : #그렇지 않으면
    print("꽝!!!")
    print("당첨 번호는",lotto,"입니다")

input()
```

실행 결과 화면

```
가지고 있는 복권 번호를 입력 : 45
꽝!!!
당첨 번호는 57 입니다
```

(가)	randint	(나)	user==lotto

2. Ex2_랜덤뺄셈문제맞추기(단, 틀렸을 경우 계속 재입력 받는다.)

코드

```
import random
print("==랜덤 뺄셈 문제 맞추기==")
num1=random.[  (가)  ](1,100) #1~100사이의 난수 발생
num2=random.randint(1,100) #1~100사이의 난수 발생
answer=num1-num2 #정답
while [  (나)  ] : #무한반복
    user=int(input(str(num1)+"-"+str(num2)+"=")) #사용자로부터 입력받기
    if([  (다)  ]) : #만약, 사용자 입력값(user)과 정답(answer)이 같으면
        print("맞았습니다.") # "맞았습니다" 출력
        [  (라)  ] #반복문 빠져나오기
    else : #그렇지 않으면
        print("틀렸습니다.") # "틀렸습니다" 출력

input()
```

실행 결과 화면

```
==랜덤 뺄셈 문제 맞추기==
74-60=14
맞았습니다.
```

```
==랜덤 뺄셈 문제 맞추기==
20-41=-11
틀렸습니다.
20-41=21
틀렸습니다.
20-41=-21
맞았습니다.
```

(가)	randint	(나)	True
(다)	user==answer	(라)	break

2019학년도 11월 실시 (정보처리와 관리) 문제해결실습 정답
2학년 ( )반 번호 : ( ) 이름 : ( )

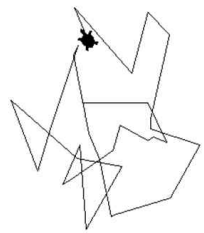
3. Ex3_랜덤으로움직이는거북이(터틀)

[조건]
1.거북이의 움직임을 30회 반복한다.
2.거북이의 움직임
  1) 1~100사이의 난수를생성하여 그 길이만큼 이동
  2) -180~180사이의 난수를생성하여 그 각도만큼 오른쪽 회전

코드

```
import turtle #turtle모듈 가져오기
import random #random모듈 가져오기

t=turtle.Turtle()
t.shape("turtle")
turtle.title("랜덤으로 움직이는 거북이")

colorList=["red","orange","yellow","green","blue","purple"]
for i in range([  (가)  ]) : # 30회 반복
    c=[  (나)  ](colorList)#colorList에서 랜덤으로 하나의 색을 추출
    t.color(c) #색 설정
    length=random.randint([  (다)  ]) #1~100사이의 난수를 생성하여 length에 저장
    t.forward(length) #거북이가 length만큼 앞으로 이동
    angle=random.randint([  (라)  ]) #-180~180사이의 난수를 생성하여 angle에 저장
    t.right(angle) #거북이가 angle만큼 오른쪽으로 회전

turtle.done()
```

실행 결과 화면



(가)	30	(나)	random.choice
(다)	1,100	(라)	-180,180

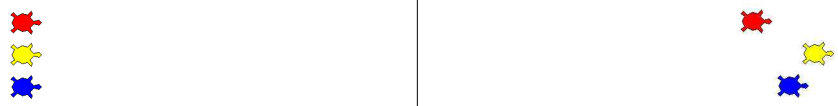
4. Ex4_터틀런(터틀)

[조건]
1.거북이의 초기 위치 : red(-300,50), yellow(-300,0), blue(-300,-50)
2.거북이들은 앞으로 가는 동작을 35번 반복한다.
단, 앞으로 갈 때 거리 값은 1~30사이의 값을 랜덤으로 발생시켜 이동함

코드

```
import turtle
import random
turtle.title("터틀런")
red=[   (가)   ]#거북이 red생성
yellow=turtle.Turtle() #거북이 yellow생성
blue=turtle.Turtle() #거북이 blue생성
#거북이 red의 설정
red.shape("turtle")
red.fillcolor("red")
red.shapesize(2)
red.penup()
red.goto(-300,50)
#거북이 yellow의 설정
yellow.shape("turtle")
yellow.fillcolor("yellow")
yellow.shapesize(2)
yellow.penup()
yellow.goto(-300,0)
#거북이 blue의 설정
blue.shape("turtle")
blue.fillcolor("blue")
blue.shapesize(2)
blue.penup()
blue.goto(-300,-50)
#거북이 경주
for i in range(35) : #35회 반복
    d1=random.randint(1,30) #1~30사이의 값 중 하나의 값을 랜덤으로 발생시킴
    red.forward([   (나)   ]) #d1만큼 앞으로 가기
    d2=random.randint(1,30) #1~30사이의 값 중 하나의 값을 랜덤으로 발생시킴
    yellow.forward([   (다)   ]) #d2만큼 앞으로 가기
    d3=random.randint(1,30) #1~30사이의 값 중 하나의 값을 랜덤으로 발생시킴
    blue.forward([   (라)   ]) #d3만큼 앞으로 가기
```

실행 결과 화면



(가)	turtle.Turtle()	(나)	d1
(다)	d2	(라)	d3

2019학년도 11월 실시 (정보처리와 관리) 문제해결실습 정답
2학년 ( )반 번호 : ( ) 이름 : ( )

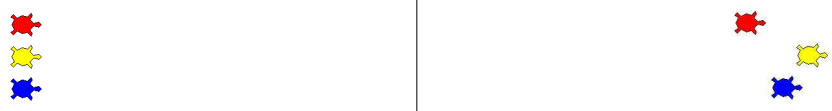
5.Ex4_터틀런(터틀)_도전 : Ex4에서 작성한 터틀런에 아래의 조건을 추가하여 1등을 출력하기

[조건]
x축의 값이 가장 큰 거북이가 1등(1등 거북이를 화면 출력)

코드

```
import turtle
import random
turtle.title("터틀런")
red=turtle.Turtle() #red생성
yellow=turtle.Turtle() #yellow생성
blue=turtle.Turtle() #blue생성
#거북이 red의 설정
red.shape("turtle")
red.color("red")
red.shapesize(2)
red.penup()
red.goto(-300,50)
#거북이 yellow의 설정
yellow.shape("turtle")
yellow.color("yellow")
yellow.shapesize(2)
yellow.penup()
yellow.goto(-300,0)
#거북이 blue의 설정
blue.shape("turtle")
blue.color("blue")
blue.shapesize(2)
blue.penup()
blue.goto(-300,-50)
#거북이 경주
for i in range([   (가)   ]) :#35회반복
    red.forward(random.randint(1,30))
    yellow.forward(random.randint(1,30))
    blue.forward(random.randint(1,30))

#거북이들의  x좌표
x1=red.[   (나)   ]  #red의 x좌표
x2=yellow.xcor() #yellow의 x좌표
x3=blue.xcor()  #blue의 x좌표를 x3

#가장 멀리 간 거북이 찾기 :x좌표가 가장 큰 값
if([   (다)   ]) : #red가 1등인 경우
    win=red
elif (x2>x1 and x2>x3) : #yellow가 1등인 경우
    [   (라)   ]
else : #blue가 1등인 경우
    win=blue

#1등 출력하기
t=turtle.Turtle() #출력을 위한 거북이
t.hideturtle()
t.penup()
t.goto(-20,180)
t.write(win.fillcolor()+"가 1등입니다"))

turtle.done()
```

실행 결과 화면



(가)	35	(나)	xcor()
(다)	x1>x2 and x1>x3	(라)	win=yellow