

소 프로젝트 1

1910896 천옥희(통계학과)

1-1

step1. 두 정수를 주었을 때 출력값

두 정수를 num1 과 num2라고 두고 생각하면, 출력값은 num1+num2이어야합니다.

step2. num1,num2의 정수 값

임의의 두 정수를 생성하라고 했으므로, random이라는 난수 모듈을 떠올렸습니다.

두 자릿수의 정수라는 조건이 있으므로, 두 자릿수의 정수인 10~99까지의 값을 넣어야 합니다. 따라서, random 모듈에서 a~b사이의 정수 난수 발생시키는 randint(a,b)를 사용해 a를 10, b를 100으로 두어 10~99까지의 수가 임의로 생성되게 하도록 하였습니다. 두 정수를 num1,num2로 변수명을 지정해주었습니다.

⇒import random이라는 난수 모듈을 사용.

```
num1=random.randint(10,100)
```

```
num2=random.randint(10,100)
```

step3. 문제 생성

문제를 생성할 때 자료에 나와 있는 예제 '문제1: 36+24?'처럼 나오기 위해서는 print('문제1: 36+24')해야겠다는 생각을 먼저 떠올렸으나, 여기에 나와 있는 36과 24의 숫자는 임의의 두 정수이므로, 위에서 미리 생성해놓은 num1과 num2를 사용해야겠다는 생각을 하였습니다. 문제를 출력하기 위해서 print('문제1: num1+num2')라는 함수를 사용하게 되면, num1과 num2는 임의의 두 자릿수의 정수가 아닌, num1과 num2로 그대로 출력되기 때문에 정수 포맷 코드인 %d를 사용했습니다.

⇒print('문제1: %d+%d' %(num1,num2))

step4. 답을 사용자 입력을 받도록 생성

위에서 만든 문제를 print 대신해서 input이라는 사용자입력을 받는 함수를 넣어줬습니다. 단, 여기서 입력을 정수로 받기 때문에 int라는 함수를 input 앞에 붙여 정수의 형태로 반환시키도록 하였습니다.

⇒int(input('문제1: %d+%d?' %(num1, num2)))

step5. 반복되는 문제1~문제5

문제1~문제5를 일일이 모두 int(input('문제1: %d+%d?' %(num1, num2))) ~ int(input('문제5: %d+%d?' %(num1,num2)))쓰기보다는 for문을 사용하여 5개의 문제를 만들면, 훨씬 간단하겠다고 생각하였습니다. 5개의 문제를 문제i 부분에서 i만을 바꿔주면 되기 때문에 range를 이용해 1~5를 포함하는 range(1,6)을 사용하면 좋을 것 같습니다. 그래서 for문을 통해 range(1,6)의 숫자를 순서대로 받아오는 i로 바꿔주었습니다. 여기서 출력을 할 때 역시, I를 그대로 넣는게 아니라, 정수 포맷형식인 %d를 사용합니다.

⇒for i in range(1,6):

```
int(input('문제%d: %d+%d?' %(i, num1, num2)))
```

step6. 정답인지 오답인지를 알려주는 프로그램 작성

step1에서 출력값은 num1+num2가 되어야 된다고 했습니다. 사용자가 답한 값이 num1과 num2의 합이 같으면, '정답입니다!'를 출력하고, 같지 않으면, '틀렸습니다!'를 출력해야합니다. 따라서, 조건문인 if문을 사용해야겠다고 생각하였습니다. 위에서 만든 int(input('문제%d: %d+%d?' %(i, num1, num2)))는 사용자가 입력한 답이므로, answer이라는 변수명을 정의해주었습니다. if문을 이용하여 answer과 num1+num2이 같다면 '정답입니다!'를 출력하고, 그렇지 않다면 '틀렸습니다!'를 출력하도록 하였습니다.

```

⇒answer=int(input('문제%d: %d+%d?' %(i, num1, num2)))
if answer==num1+num2:
    print('정답입니다!')
else:
    print('틀렸습니다!')

```

step7. 위 과정 순서정리

첫 번째, 임의의 두 정수를 생성하기 위해 필요한 import random이라는 모듈을 지정해줍니다. 두 번째로 for 문을 써주고 그 밑에 step2에서 정리한 num1과 num2를 써줍니다. num1과 num2를 for문 전에 쓰게 되면, 모든 문제의 num1과 num2의 값이 같아지기 때문에 for문 먼저 쓰고, num1과 num2를 지정해줍니다. 세 번째로 step3와 step4를 통해 생성한 문제를 입력해줍니다. 이때, answer이라는 변수명으로 지정해주고 써줍니다. 그 이유는, input안에 문제를 넣었으므로, 사용자가 입력한 답을 answer이라고 지정해 주기 위해서입니다. 마지막으로, step6에서 정리한 if 문을 써 정답과 오답을 안내해주는 프로그램을 완성해줍니다. 이 내용을 정리하면 사진 1-1(1)과 같이 정리할 수 있으며, 이를 실행시킨 결과는 사진 1-1(2)과 같습니다.

<pre> #1-1 import random #난수 발생모듈, 임의의 두 정수를 생성하기 위해 필요 for i in range(1,6): #문제1~문제5 5개의 문제 출제를 위한 함수 num1=random.randint(10,100) #첫번째 정수의 변수, 임의의 두자리수의 정수 생성 num2=random.randint(10,100) #두번째 정수의 변수, 임의의 두자리수의 정수 생성 answer=int(input('문제%d: %d+%d?' %(i, num1, num2))) #사용자가 입력한 답, 사용자의 답 입력 받기 if answer==num1+num2: #사용자가 입력한 답이 정답일때의 '정답입니다!'를 출력 print('정답입니다!') else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다!'를 출력 print('틀렸습니다!') </pre>	<pre> 문제1: 55+24?79 정답입니다! 문제2: 25+18?3 틀렸습니다! 문제3: 33+89?122 정답입니다! 문제4: 37+16?5 틀렸습니다! 문제5: 68+71?139 정답입니다! </pre>
---	---

사진1-1(1)

사진1-1(2)

1-2

step1. 1-1의 내용을 그대로 가져오고, 문제에서 다르게 요구한 점을 찾기.

차이점=연산자가 덧셈, 뺄셈, 곱셈을 랜덤하게 가져오라는 것, 연산자에 정답이 달라지는 것.

step2. 연산자 생성

연산자를 리스트를 만들어 먼저 생성하였습니다. 리스트 안에 연산자를 그냥 넣게 되면, 구문오류가 나게 됩니다. 그러므로 각각의 연산자를 작은따옴표('')나, 큰따옴표("")로 묶어야 됩니다. 이 연산자 리스트를 symbol이라는 변수명을 주었습니다.

```
⇒symbol = ['+', '-', '*']
```

step3. 연산자 랜덤 생성

step2에서 연산자를 생성했지만, 문제에서 요구한 것은 랜덤으로 정해지도록 하는 것입니다. 위에서 생성한 symbol 연산자리스트에서 랜덤으로 선택할 수 있도록 해야겠다고 생각했습니다. 그래서 random 모듈의 choice함수를 사용해 연산자 리스트에서 랜덤으로 연산자가 정해질 수 있도록 하였습니다.

```
⇒random.choice(symbol)
```

step4. 문제 수정

1-1의 문제에서는 연산자가 +밖에 없었기 때문에 문제 안에 '+'를 넣어줬습니다. 1-2문제에서는 랜덤으로 뽑힌 연산자에 맞게 문제를 바꿔줘야 합니다. 먼저, 위에서 생성한 랜덤 연산자를 arithmetic으로 변수명을 지정해주었습니다. 1-1에서 생성한 문제에서 '+'를 arithmetic으로 바꿔 줘야한다는 생각을 했습니다. 위에서 정수 포맷코드를 사용했기 때문에, 여기서도 문자열 포맷코드를 사용했습니다. 문자열 포맷코드인 %s를 '+'자리에 넣어주고, 뒤에 %artithmetic을 넣어줬습니다.

```
⇒arithmetic=random.choice(symbol)
```

```
answer=int(input('문제%d: %d%s%d?'%(i, num1, arithmetic, num2)))
```

step5. 정답인지 오답인지를 알려주는 프로그램 수정

위에서는 모든 연산자가 '+'이었기 때문에 if문과 else문 하나씩으로 끝이 났습니다. 하지만, 여기서는 경우의 수가 3가지가 있습니다. arithmetic이 '+'일 경우, '-'일 경우, '*'일 경우 총 3가지 경우를 따져봐야 합니다. '+' 경우에는 answer==num1+num2를 그대로 사용합니다. '-' 경우에는 answer==num1-num2로 바꾸주고, '*' 경우에는 answer==num1*num2로 바꾸주었습니다. 이를 작성하기 위해서는 if, else, elif조건문을 이용하여 '+'일 경우, answer==num1+num2이면, '정답입니다!' 그렇지 않으면, '틀렸습니다!'를 작성하고, '-'일 경우, answer ==num1-num2이면, '정답입니다!' 그렇지 않으면, '틀렸습니다!'라고 작성을 했습니다. 마지막으로, '*'일 경우, answer==num1*num2이면, '정답입니다!' 그렇지 않으면, '틀렸습니다!'로 작성을 하였습니다.

```
⇒if arithmetic=='+':
```

```
    if answer==num1+num2:
        print('정답입니다!')
```

```
    else:
        print('틀렸습니다!')
```

```
elif arithmetic=='-':
```

```
    if answer==num1-num2:
        print('정답입니다!')
```

```
    else:
        print('틀렸습니다!')
```

```
if arithmetic=='*':
```

```
    if answer==num1*num2:
        print('정답입니다!')
```

```
    else:
        print('틀렸습니다!')
```

step6. 위 과정 순서정리

순서는 1-1에서와 거의 똑같습니다. 1-1의 순서에서 내용을 추가하고, 수정을 해줘야합니다. 첫 번째로, symbol 연산자 리스트와 arithmetic 랜덤 연산자 생성함수를 for문 안에 넣어줍니다. 1-1의 step7에서 num1가 num2를 for문 안에 써줘야 하는 이유를 말했듯이 arithmetic도 역시 for 문 안에 써 줘야합니다. 밖에 써주게 되면, 모든 문제의 연산자는 같게 되기 때문입니다. 다음으로, 1-1의 정답과 오답을 알려주는 if문을 1-2에서는 각각의 연산자에 맞게 바꿔줘야하기 때문에 step5에서 만든 조건문으로 바꿔 줍니다. 이렇게 순서대로 정리를 하게 되면 사진 1-2(1)처럼 정리할 수 있으며, 실행을 하게 되면 사진 1-2(2)과 같습니다.

```
#1-2
import random #난수 발생모듈, 임의의 두 정수와 연산자를 생성하기 위해 필요

for i in range(1,6): #문제1~문제5 5개의 문제 출제
    num1=random.randint(10,100) #첫번째 정수의 변수, 임의의 두자리수의 정수 생성
    num2=random.randint(10,100) #두번째 정수의 변수, 임의의 두자리수의 정수 생성
    symbol = ['+', '-', '*'] # 연산자 리스트
    arithmetic = random.choice(symbol) #랜덤하게 정해진 연산자의 변수, 연산자를 랜덤생성
    answer=int(input('문제%d: %d%s%d?'%(i, num1, arithmetic, num2))) #사용자가 입력한 답, 사용자의 답 입력 받기
    if arithmetic=='+': #랜덤하게 정해진 연산자가 +일 경우의 조건문
        if answer==num1+num2: #사용자가 입력한 답이 정답일때의 '정답입니다!'를 출력
            print('정답입니다!')
        else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다!'를 출력
            print('틀렸습니다!')
    elif arithmetic=='-': #랜덤하게 정해진 연산자가 -일 경우의 조건문
        if answer==num1-num2: #사용자가 입력한 답이 정답일때의 '정답입니다!'를 출력
            print('정답입니다!')
        else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다!'를 출력
            print('틀렸습니다!')
    else: #랜덤하게 정해진 연산자가 *일 경우의 조건문
        if answer==num1*num2: #사용자가 입력한 답이 정답일때의 '정답입니다!'를 출력
            print('정답입니다!')
        else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다!'를 출력
            print('틀렸습니다!')
```

문제1: 81+25?**106**
정답입니다!
문제2: 86-86?**0**
정답입니다!
문제3: 26-38?**2**
틀렸습니다!
문제4: 51+33?**3**
틀렸습니다!
문제5: 97*84?**8148**
정답입니다!

사진 1-2(1)

사진 1-2(2)

1-3

step1. 1-2를 그대로 가져온 후, 문제에서 다르게 요구한 점 찾기

차이점: 5초라는 시간제한기능을 주는 것과 5초를 초과하면, 답을 맞췄지만 시간이 초과되었음을 안내하고, 틀린 것으로 처리하는 것, 5문제 중 맞춘 문제의 개수 출력하는 것

step2. 문제를 맞추는데 걸린 시간 계산

문제의 답을 입력한 시간에서 문제를 낸 시간을 빼주면 문제를 맞추는데 걸리는 시간을 구할 수 있습니다. 따라서 문제를 낸 시간을 starttime이라는 변수명을 주고, time모듈의 time을 통해 문제를 낸 시간을 받습니다. 문제의 답을 입력한 시간을 endtime이라는 변수명을 주고, time모듈의 time을 이용해 문제를 입력한 시간을 받습니다. 문제를 낸 시간을 받기 위해서 문제를 내기 직전에 starttime을 써주고, 문제의 답을 입력한 시간을 받기 위해서는 문제를 낸 직후에 endtime을 써주면 문제를 낸 시간과 문제의 답을 입력한 시간을 받을 수 있습니다. 여기서 얻고자 한 것은 문제를 맞추는데 걸리는 시간이므로, endtime에서 starttime을 뺀 시간을 taketime이라는 변수명을 주고 받아옵니다.

```
⇒starttime=time.time()
```

```
answer=int(input('문제%d: %d%s%d?'%(i, num1, arithmetic, num2)))
```

```
endtime=time.time()
```

```
taketime=endtime-starttime
```

step3. 5초의 시간제한

5초의 시간제한을 주기위해서 위에서 구한 taketime함수를 이용합니다. taketime은 문제를 맞추는데 걸린 시간임으로 걸린 시간이 5초를 넘지 않으면 된다는 조건을 주면 됩니다. 따라서, 1-2의 step5에서 만든 조건문을 수정해주면 됩니다. 답을 맞추고, taketime이 5초를 넘지 않으면 '시간 내에 맞췄습니다.'를 출력해주고, 그렇지 않으면, '답을 맞췄으나 시간이 초과되었습니다.'를 출력해줍니다. 답이 틀렸으면, '틀렸습니다.'를 출력해줍니다. 이를 위해 arithmetic이 '+'일 때, answer==num1+num2이고, taketime<5 이면, '시간 내에 맞췄습니다.'를 print해줍니다. 만약, taketime<5가 아니면, '답을 맞췄으나 시간이 초과되었습니다.'를 print해줍니다. 여기서 answer==num1+num2가 아니게 되면, '틀렸습니다.'를 print해줬습니다.

```
⇒if arithmetic == '+':
```

```
    if answer==num1+num2:
```

```
        if taketime<5:
```

```
            print('시간 내에 맞췄습니다.')
```

```
        else:
```

```
            print('답을 맞췄으나 시간이 초과되었습니다.')
```

```
    else:
```

```
        print('틀렸습니다')
```

arithmetic이 '-'와 '*'일 때, if arithmetic=='-':

```
    if answer==num1-num2:
```

if arithmetic=='*':

```
    if answer==num1*num2
```

여기만 다르고 그 이후부터는 위의 arithmetic=='+'의 코드와 같게 해줬습니다. 이렇게 하면 5초라는 제한시간을 조건문을 통해 줄 수 있게 됩니다.

```
⇒if arithmetic == '-':
```

```
    if answer==num1-num2:
```

```
        if taketime<5:
```

```
            print('시간 내에 맞췄습니다.')
```

```
        else:
```

```
            print('답을 맞췄으나 시간이 초과되었습니다.')
```

```
    else:
```

```
        print('틀렸습니다')
```

```

if arithmetic == '*':
    if answer==num1*num2:
        if taketime<5:
            print('시간 내에 맞했습니다.')
        else:
            print('답을 맞했으나 시간이 초과되었습니다.')
    else:
        print('틀렸습니다')

```

step4. 맞춘 문제의 개수

시간 내에 맞춘 문제의 개수를 correct라는 변수명을 주고, 0이라는 초깃값을 주었습니다. 문제를 처음 시작했을 때, 맞춘 문제 수는 아무 것도 없기 때문에 0이라는 초깃값을 줍니다. 시간 내에 문제를 맞추게 되면, correct의 값이 1개씩 올라가게 만들어야 함으로, correct=correct+1이라는 함수를 써 줬습니다. 이를 step3에서의 조건문 안에 넣어 줄 것인데, if take<5:이라는 조건문 밑에다가 넣어줍니다. 이 조건문은 시간 내에 문제를 맞췄을 경우이기 때문에 correct의 값이 +1이 되어야함으로 print('시간 내에 맞했습니다.')라는 출력함수 밑에 넣어줍니다.(print('시간 내에 맞했습니다.')위에 써도 됩니다. if take<5밑에만 써주면 됩니다.) 이것은 arithmetic='+'일 경우, '-'일 경우, '*' 경우 모든 경우에 포함되기 때문에 각각의 if taketime<5:라는 조건문 밑에 넣어주었습니다. 5개의 문제를 모두 풀게 되면, 시간 내에 맞춘 문제의 개수가 correct의 값에 저장되기 때문에, 이를 출력해 몇 개를 맞췄는지 사용자가 알 수 있도록 해줘야합니다. 따라서 print('정답을 맞춘 개수: %d/5' %correct)를 제일 마지막에 써줍니다. 여기서 %d를 사용한 이유는 작은따옴표 안에 correct를 넣게 되면, 출력된 내용이 '정답을 맞춘 개수: correct/5'이렇게 correct의 값이 아닌 correct가 그대로 출력해 나오기 때문에 정수 포맷코드를 넣어주어 correct의 값이 출력될 수 있도록 만들었습니다.

⇒correct=0

```

if arithmetic == '+':
    if answer==num1+num2:
        if taketime<5:
            print('시간 내에 맞했습니다.')
            correct=correct+1
            ...

if arithmetic == '-':
    if answer==num1-num2:
        if taketime<5:
            print('시간 내에 맞했습니다.')
            correct=correct+1
            ...

if arithmetic == '*':
    if answer==num1*num2:
        if taketime<5:
            print('시간 내에 맞했습니다.')
            correct=correct+1
            ...

```

```

print('정답을 맞춘 개수: %d/5' %correct)

```


step5. 위과정 순서정리

1-2의 순서와 비슷하지만, 1-2의 순서를 추가하고 수정해야할 것들이 있습니다. 첫 번째로, 시간제한을 주기위해 time 모듈을 사용했기 때문에 import random 밑에 import time을 추가 해줘야합니다. 두 번째, 맞춘 개수를 출력하기 위해서 correct=0으로 초깃값을 설정해 줘야합니다. 이것은 for문 전에 입력을 해줘야지만 for문 안에서 정답을 맞췄을 경우 1씩 증가합니다. 만약, for문 안에 쓰게 되면, 문제1을 실행할 때 correct=0이고, 문제를 시간 내에 맞추게 되면 correct=1이 될 것입니다. 하지만, 문제2를 실행하게 되면, 다시 correct=0으로 설정되어 시간 내에 문제1을 맞추고, 문제2를 맞춰도 correct=2가 아닌 correct=1이 되어버립니다. 그렇기 때문에 correct=0은 for문 전에 정의를 해줘야 합니다. 세 번째는 for문 안에서 step2에서 정리한대로 answer위에 starttime, 밑에는 endtime과 taketime을 추가해 줍니다. 네 번째로는 step3에서 정리한대로, 5초의 시간제한을 주는 조건문을 각각의 arithmetic 밑에 수정해줍니다. 마지막으로 step4에서 정의한 correct=correct+1를 if taketime<5:라는 조건문 밑에 추가해줍니다. 그리고 print('정답을 맞춘 개수: %d/5' %correct)를 for 문에서 나와 제일 마지막에 작성해줍니다. for문 안에 작성을 하게 되면, 문제1, 문제2, 문제3, 문제4, 문제5를 풀 때마다 출력되기 때문에 for 문을 벗어나 작성을 해줘 마지막에 한 번만 나올 수 있도록 해줬습니다. 이것을 사진1-3(1)과 같이 정리할 수 있으며, 이를 실행한 내용은 사진1-3(2)와 같습니다.

```
#1-3
import random #난수 발생모듈, 임의의 두 정수와 연산자를 생성하기 위해 필요
import time #시간과 관련된 모듈, 시간 제한을 주기위해 필요

correct= 0 #정답개수의 초깃값

for i in range(1,6): #문제1~문제5 5개의 문제 출제
    num1=random.randint(10,100) #첫번째 정수의 변수, 임의의 두자릿수의 정수 생성
    num2=random.randint(10,100) #두번째 정수의 변수, 임의의 두자릿수의 정수 생성
    symbol = ['+', '-', '*'] #연산자 리스트
    arithmetic = random.choice(symbol) #랜덤하게 정해진 연산자의 변수, 연산자 랜덤생성
    starttime=time.time() #문제출제시간
    answer=int(input('문제%d: %d%s%d?'%(i, num1, arithmetic, num2))) #사용자가 입력한 답, 사용자의 답 입력 받기
    endtime=time.time() #문제맞춘시간
    taketime=endtime-starttime #문제 맞추는데 걸린 시간, 문제맞춘시간-문제출제시간
    if arithmetic == '+': #랜덤하게 정해진 연산자가 +일 경우의 조건문
        if answer==num1+num2: #두개의 정수를 더한 값을 맞췄을 경우
            if taketime < 5: #시간 내에 맞췄을 경우 '시간내에 맞췄습니다'를 출력
                print('시간 내에 맞췄습니다.')
                correct= correct+1 #시간 내에 맞췄을 때 정답의 갯수 1증가
            else: #맞췄지만 시간을 초과했을 때 '답을 맞췄으나 시간이 초과되었습니다'를 출력
                print('답을 맞췄으나 시간이 초과되었습니다.')
        else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다.'를 출력
            print('틀렸습니다.')
    elif arithmetic == '-': #랜덤하게 정해진 연산자가 -일 경우의 조건문
        if answer==num1-num2: #두개의 정수를 뺀 값을 맞췄을 경우
            if taketime < 5: #시간 내에 맞췄을 경우 '시간내에 맞췄습니다'를 출력
                print('시간 내에 맞췄습니다.')
                correct= correct+1 #시간 내에 맞췄을 때 정답의 갯수 1증가
            else: #맞췄지만 시간을 초과했을 때 '답을 맞췄으나 시간이 초과되었습니다'를 출력
                print('답을 맞췄으나 시간이 초과되었습니다.')
        else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다.'를 출력
            print('틀렸습니다.')
    else: #랜덤하게 정해진 연산자가 *일 경우의 조건문
        if answer==num1*num2: #두개의 정수를 곱한값을 맞췄을 경우
            if taketime < 5: #시간 내에 맞췄을 경우 '시간내에 맞췄습니다'를 출력
                print('시간 내에 맞췄습니다.')
                correct= correct+1 #시간 내에 맞췄을 때 정답의 갯수 1증가
            else: #맞췄지만 시간을 초과했을 때 '답을 맞췄으나 시간이 초과되었습니다'를 출력
                print('답을 맞췄으나 시간이 초과되었습니다.')
        else: #사용자가 입력한 답이 틀렸을 때의 '틀렸습니다.'를 출력
            print('틀렸습니다.')
print('정답을 맞춘 갯수: %d/5' %correct) #5개 중에 시간내에 맞춘 정답의 개수를 출력
```

문제1: 24*79? 1896
답을 맞췄으나 시간이 초과되었습니다.
문제2: 44+52? 96
시간 내에 맞췄습니다.
문제3: 68-89? -21
답을 맞췄으나 시간이 초과되었습니다.
문제4: 11*17? 187
틀렸습니다.
문제5: 48-12? 36
시간 내에 맞췄습니다.
정답을 맞춘 갯수: 2/5

사진1-3(2)

사진 1-3(1)