|  |
| --- |
| **R E P O R T** |

|  |
| --- |
| **[ 컴퓨팅 사고]** |



|  |  |
| --- | --- |
| **학 과** | **> 컴퓨터공학부** |
| **교수님** | **> 김종남교수님** |
| **학 번** | **> 201911608** |
| **이 름** | **> 김 지 환** |
| **제출일** | **> 2022.10.22** |



**목 차**

**과제 1 - 1장 프로그래밍과 파이썬 소개 연습문제** **……… 03p**

**과제 2 – 2장 변수와 수식 연습문제 …..…………………… 05p**

**과제 3 - 3장 조건문 연습문제 ………………………………… 10p**

**과제 4 - 4장 반복문 연습문제 ………………………………… 15p**

**과제 5 – 컴퓨터 조립 시 필요한 부품 선정 ……………… 19p**

**과제 6 - 5장 함수 연습문제 …………………………………… 21p**

**과제 7 - 6장 자료구조(리스트) 연습문제 ..……………… 26p**

**과제 1 - 1장 프로그래밍과 파이썬 소개 연습문제**

02. 다음과 같이 사각형 안에 이름을 출력하는 프로그램을 작성해보자.

ex) 실행결과

+================+  
| 홍길동 |  
+================+

|  |
| --- |
| =======Source Code======= |
| print("+================+") print("| 홍길동 |") print("+================+") |
| ========실행결과========= |
|  |

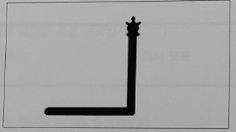
04. 다음과 같은 결과를 출력하는 프로그램을 작성해보자. 6은 1+2+3을 계산하여 출력한다.

ex) 실행결과

1+2+3을 계산하면 6입니다.

|  |
| --- |
| =======Source Code======= |
| print(f"1+2+3을 계산하면 {1+2+3}입니다.") |
| ========실행결과========= |
|  |

07. 터틀 그래픽에서 width() 함수를 호출하면 거북이가 그리는 선의 두께를 두껍게 한다. 거북이를 이동하여서 다음과 같이 두께가 10인 선을 그려보자.



|  |
| --- |
| =======Source Code======= |
| import turtle t = turtle.Turtle() turtle.setup(width = 500, height = 300) t.shape("turtle") t.penup() t.goto(-150,-100) t.pendown() t.width(10) t.forward(180) t.left(90) t.forward(200) input() |
| ========실행결과========= |
|  |

**과제 2 - 2장 변수와 수식 연습문제**

02. 농장에 2마리의 닭, 4마리의 돼지, 3마리의 소가 있다고 하자. 전체 다리의 수는 몇 개 일까? 계산하는 프로그램을 작성하고 테스트해보자.

ex) 실행 결과

닭의 수: 2

돼지의 수: 3

소의 수: 4

전체 다리의 수: 32

|  |
| --- |
| =======Source Code======= |
| animal = {"닭의 수" : 2, "돼지의 수" : 3, "소의 수" : 4} leg = 0 for type in animal:  print(f"{type}: {animal[type]}")  if type == "닭의 수":  leg += animal[type] \* 2  continue  leg += animal[type]\*4 print() print(f"전체 다리의 수 : {leg}") |
| ========실행결과========= |
|  |

04. 사용자로부터 시간과 분을 받아서 초로 변환하는 프로그램을 작성해보자.

ex) 실행결과

시간을 입력하시오: 1

분을 입력하시오: 3

1 시간 3 분은 3780 초입니다.

|  |
| --- |
| =======Source Code======= |
| hour, min = 0, 0 hour = int(input("시간을 입력하시오: ")) min = int(input("분을 입력하시오: ")) sec = hour\*3600 + min\*60 print(f"{hour} 시간 {min} 분은 {sec} 초입니다.") |
| ========실행결과========= |
|  |

07. 사용자로부터 두 점의 좌표 (x1, y1)과 (x2, y2)를 입력받아서 두 점 사이의 거리를 계산하는 프로그램을 작성해보자. 스크립트 모드로 작성하라. 거리는 다음 식으로 계산한다.

ex) 실행결과

x1=0

y1=0

x2=100

y2=100

두 점 사이의 거리= 141.4213562373095

|  |
| --- |
| =======Source Code======= |
| d = {"x1":0,"y1":0,"x2":0,"y2":0} for i in d:  d[i] = int(input(i+"=")) dx, dy = (d["x1"]-d["x2"])\*\*2, (d["y1"]-d["y2"])\*\*2 distance = (dx+dy)\*\*0.5 print(f"두 점 사이의 거리= {distance}") |
| ========실행결과========= |
|  |

09. 사용자로부터 4자리의 정수를 받아서 자리수의 합을 계산하는 프로그램을 작성하여 보자. 예를 들어서 사용자가 1234를 입력하였다면 1+2+3+4를 계산하면 된다. 나머지 연산자와 정수 나눗셈 연산자 //를 적극적으로 사용해보자.

ex) 실행결과

정수 = 1234

10

|  |
| --- |
| =======Source Code======= |
| n = int(input("정수=")) num, mod, n = 0, n%10, n//10 num, mod, n = num+mod, n%10, n//10 num, mod, n = num+mod, n%10, n//10 num, mod, n = num+mod, n%10, n//10 print(num+mod) |
| ========실행결과========= |
|  |

11. 사용자로부터 파일이 위치한 드라이브 이름(c), 디렉토리 이름(\test\), 파일이름(sample), 확장자(py)를 받아서 완전한 파일 이름(c:\test\sample.py)으로 만드는 프로그램을 작성해보자.

ex) 실행결과

드라이브 이름: c

디렉토리 이름: \test\

파일 이름: sample

확장자: py

완전한 이름은 c:\test\sample.py

|  |
| --- |
| =======Source Code======= |
| path = input("드라이브 이름: ")+":" path += input("디렉토리 이름: ") path += input("파일 이름: ") path += "."+input("확장자: ") print(path) |
| ========실행결과========= |
|  |

**과제 3 - 3장 조건문 연습문제**

03. 사용자로부터 하나의 문자를 입력받아서 문자가 ‘R’이나 ‘r’이면 “Rectangle”이라고 출력한다. “T”이거나 ‘t’이면 “Triangle”, ‘C’ 이거나 ‘c’이면 “Circle”이라고 출력하는 프로그램을 작성한다. 그 외의 문자가 들어오면 “Unknown”이라고 출력한다.

Ex) 실행결과

문자를 입력하시오: C

Circle

|  |
| --- |
| =======Source Code======= |
| def check(c):  shape = ["Rectangle", "Triangle", "Circle"]  for char in shape:  if char[0] == c:  return char  return "Unknown"  char = input("문자를 입력하시오: ").upper() print(check(char)) |
| ========실행결과========= |
|  |

06. 컴퓨터와 가위, 바위, 보 게임을 하는 프로그램을 작성하라. 컴퓨터는 사용자에게 알리지 않고 가위, 바위, 보 중에서 임의로 하나를 선택한다. 사용자는 프로그램의 입력 안내 메시지에 따라서, 3개 중에서 하나를 선택하게 된다. 사용자의 선택이 끝나면 컴퓨터는 누가 무엇을 선태하였고 누가 이겼는지, 비겼는지를 알려준다.

Ex) 실행결과

선택하시오(1: 가위 2: 바위 3: 보) 1

컴퓨터의 선택(1: 가위 2: 바위 3:보) 2

컴퓨터가 이겼음

|  |
| --- |
| =======Source Code======= |
| import random me = input("선택하시오(1: 가위 2: 바위 3: 보) ") me, computer = int(me), random.randint(1, 3)  print("컴퓨터의 선택(1: 가위 2: 바위 3: 보)", computer)  if me == computer: print("비겼음") elif computer == 3 and me == 1: print("내가 이겼음") elif computer == 1 and me == 3: print("컴퓨터가 이겼음") elif computer < me : print("내가 이겼음") else : print("컴퓨터가 이겼음") |
| ========실행결과========= |
| . |

09. 본문에서 덧셈 퀴즈를 자동으로 생성해보았다. 이번에는 덧셈, 뺄셈, 곱셈, 나눗셈 중에서 하나를 랜덤하게 선택하고 피연산자도 난수로 생성하여 사용자에게 제시하고 사용자의 답을 자동으로 채점하는 프로그램을 작성해보자.

Ex) 실행결과

1 / 5의 값은?0.2

맞았습니다.

7+5의 값은?12

맞았습니다.

|  |
| --- |
| =======Source Code======= |
| import random a = random.randint(1,9) b = random.randint(1,9) c = random.randint(0,3) operator = ["+","-","/","\*"] result = input(f"{a} {operator[c]} {b}의 값은?") if c==0 and str(a+b) == result:  print("맞았습니다.") elif c==1 and str(a-b) == result:  print("맞았습니다.") elif c==2 and str(a/b) == result:  print("맞았습니다.") elif c==3 and str(a\*b) == result:  print("맞았습니다.") else:  print("틀렸습니다.") |
| ========실행결과========= |
|  |

14. 사용자가 2차 방정식의 각 계수를 입력하면 2차 방정식의 근을 계산하여서 출력하는 프로그램을 작성한다. 2차 방정식의 판별식을 먼저 계산하여서 하나의 실근을 가지는지, 2개의 실근을 갖는지, 실근이 존재하지 않는지를 먼저 출력하라.

Ex) 실행결과

a를 입력하시오: 2

b를 입력하시오: -7

c를 입력하시오: 5

실근은 1.000000과 2.500000입니다.

|  |
| --- |
| =======Source Code======= |
| a = int(input("a를 입력하시오: ")) b = int(input("b를 입력하시오: ")) c = int(input("c를 입력하시오: ")) D = (b\*\*2) - (4\*a\*c) if D > 0:  D = D\*\*0.5  d1, d2 = (0-b-D)/(2\*a), (0-b+D)/(2\*a)  print(f"실근은 {d1}과 {d2}입니다.") elif D == 0:  print(f"중근은 {0-(b/2\*a)}입니다.") else:  print("허근입니다.") |
| ========실행결과========= |
|  |

16. 사용자로부터 pH농도를 받아서 “알칼리”, “산”, “중성”으로 나누어서 출력하는 프로그램을 작성하라.

Ex) 실행결과

pH를 입력하시오: 9.0

알칼리입니다.

|  |
| --- |
| =======Source Code======= |
| pH = float(input("pH를 입력하시오: ")) if pH<0 or pH>14:  print("pH를 제대로 입력하세요")  exit() print("중성입니다."if pH==7.0 else("알칼리입니다."if pH>7.0 else"산성입니다.")) |
| ========실행결과========= |
|  |

**과제 4 - 4장 반복문 연습문제**

02. 다음 리스트에 저장된 정수들의 합을 계산하는 프로그램을 작성해보자. 내장 함수 sum()은 이용하지 않는다.

myLisy = [ 11, 22, 23, 99, 81, 93, 35 ]

Ex) 실행결과

정수들의 합은 364

|  |
| --- |
| =======Source Code======= |
| myLisy, add = [ 11, 22, 23, 99, 81, 93, 35 ], 0 for i in myLisy:  add+=i print("정수들의 합은",add) |
| ========실행결과========= |
|  |

05. 중첩 반복문을 사용하여서 다음과 같이 출력하는 프로그램을 작성하여 보자.

Ex) 실행결과

정수를 입력하시오: 5

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

|  |
| --- |
| =======Source Code======= |
| n = input("정수를 입력하시오: ") for i in range(1,int(n)+1):  for j in range(1, i+1):  print(j, end=" ")  print() |
| ========실행결과========= |
|  |

11. 사용자가 2개의 정수(a와 b)를 입력하면 이들 정수의 최대 공약수를 찾아보자. 예를 들어서 3과 6의 최대 공약수는 3이다. 최대 공약수를 계산하는 유클리드의 알고리즘이 있지만 우리는 아주 직관적인 방법을 사용해보자. 정수 k를 1부터 하나씩 증가시키면서 a%k==0이고 b%k==0인 정수를 찾는다. 이들 정수는 모두 약수이고 이 중에서 최대 정수가 최대 공약수가 된다.

Ex) 실행결과

첫 번째 정수를 입력하시오: 3

두 번째 정수를 입력하시오: 6

3과 6의 최대 공약수는 3입니다.

|  |
| --- |
| =======Source Code======= |
| a = input("첫 번째 정수를 입력하시오: ") b = input("두 번째 정수를 입력하시오: ") a, b, gcd = int(a), int(b), 0 for k in range(1, min(int(a),int(b))+1):  if a % k == 0 and b % k == 0:  gcd = k print(f"{a}와 {b}의 최대 공약수는 {gcd}입니다.") |
| ========실행결과========= |
|  |

14. 2와 20 사이에 있는 모든 소수(prime number)를 찾는 프로그램을 작성하라. 정수가 소수가 되려면 1과 자기 자신만을 약수로 가져야 한다.

Ex) 실행결과

2 3 5 7 11 13 17 19

|  |
| --- |
| =======Source Code======= |
| for i in range(2, 21):  prime = 1  for j in range(2, int(i\*\*0.5)+1):  if i % j == 0:  prime = 0  break  if prime : print(i, end=" ") |
| ========실행결과========= |
|  |

17. 0부터 9까지의 정수에 대하여 반복한다. 특정 숫자가 3으로 나누어 떨어지면 ‘fizz’를 출력하고, 5로 나누어 떨어지면 ‘buzz’를 출력한다. 3과 5로 나누어 떨어지면 ‘fizzbuzz’를 출력한다. 숫자가 3또는 5로 나누어 떨어지지 않으면 “\*’을 출력하라.

Ex) 실행결과

fizzbuzz

\*

\*

fizz

\*

buzz

fizz

\*

\*

fizz

|  |
| --- |
| =======Source Code======= |
| for i in range(10):  if not i%3 and not i%5:  print("fizzbuzz")  elif not i%3:  print("fizz")  elif not i%5:  print("buzz")  else:  print("\*") |
| ========실행결과========= |
|  |

**과제5 – 조사**

* **업무의 목적 (예: 게임, 사무/가정용, 연구개발 등)**

|  |
| --- |
| 컴퓨터를 이용해서 주로 프로젝트나 개인 공부를 하거나 게임을 하므로 업무의 목적으로 게임을 선정했다. 주로하는 게임은 LeagueOfLegends이다. |

* **컴퓨터 조립을 위해 필요한 부품을 선정**

|  |
| --- |
| MSI에서 PC의 구성을 위해서 반드시 포함해야 하는 필수 컴퓨터 부품을 소개했다. 해당 부품들을 기준으로 찾아보려고 한다.  MSI에서 소개하는 필수 컴퓨터 부품   * CPU, CPU쿨러, 마더보드, 메모리, 그래픽 카드, 하드 드라이브, 전원공급장치, 케이스, 주변장치 * <https://kr.msi.com/Landing/how-to-build-a-pc> (참고) |
| LeagueOfLegends 에서 권장하는 시스템 사양이 있다. 이것을 바탕으로 컴퓨터 부품을 선정    [https://download.kr.riotgames.com/league?\_gl=1\*5eccp6\*\_ga\*MTMwMzU0MDMxMS4xNjY2MzgwODMy\*\_ga\_FXBJE5DEDD\*MTY2NjM4MDgzMS4xLjAuMTY2NjM4MDgzNC41Ny4wLjA](https://download.kr.riotgames.com/league?_gl=1*5eccp6*_ga*MTMwMzU0MDMxMS4xNjY2MzgwODMy*_ga_FXBJE5DEDD*MTY2NjM4MDgzMS4xLjAuMTY2NjM4MDgzNC41Ny4wLjA). (참고)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | CPU | CPU쿨러 | 마더보드 | 메모리 | HDD | 그래픽 | power | | I5-3300↑ | 공랭식 |  | 4GB↑ | SSD | Geforce 560 ↑ |  |  * 케이스는 디자인용, 주변 장치는 입출력 장치이므로 제외함   정리된 내용을 바탕으로 아래와 같이 선정 ( 최고 사양을 생각해 선정 )   |  |  | | --- | --- | | CPU | AMD 라이젠3-3세대 3300X | | 메모리 | 마이크론 Crucial DDR4 8G PC4-25600 CL22 | | 마더보드 | GIGABYTE B450M AORUS ELITE | | 그래픽카드 | PALIT 지포스 GTX 1660 Dual NL D5 6GB | | SSD | WD BLUE 3D SSD 500GB | | POWER | Micronics COOLMAX FOCUS 500W 80Plus 230V EU |  * 라이젠 CPU 에서 쿨러가 포함되어 있음 |

* **각 부품들의 주요 규격과 사진**

|  |  |
| --- | --- |
|  | CPU  AMD 라이젠3-3세대 3300X는 주요 구격으로  PCIe 4.0까지 지원하고 메모리 규격은 DDR4,  쿨러의 경우 규격에 맞는 전용 쿨러가 있음.  전력 65W |
|  | 마더보드  확장 슬롯 : PCIe 3 지원 CPU 규격 내 – 호환  메모리 규격 : DDR 4 지원 – CPU 규격과 동일 - 호환  그래픽 규격 : Quad Gpu 및 2-way 그래픽기술 지원  CPU 규격 : AMD 라이젠 3세대 지원 - 호환 |
|  | 그래픽카드  PCIe 3.0 지원 – CPU, 마더보드 규격과 호환  권장 시스템 전력 300W  메모리 용량 4GB – 메모리 충분 |
|  | 메모리  CPU, 마더보드와 규격 호환됨, 그래픽카드 메모리 충분  전원 1.2V |
|  | SSD  PCIe 3.0, 전력 4.5W |
|  | 500W , 230V로 충분 |

**과제 6 - 5장 함수 연습문제**

03. 2번에서는 4개의 함수를 작성하였다. 하나의 함수 calc() 안에서 덧셈, 뺄셈, 곱셈, 나눗셈을 모두 수행하고 4개의 계산값을 동시에 반환하도록 수정해보자.

Ex) 실행결과

첫 번째 정수를 입력하시오: 10

두 번째 정수를 입력하시오: 20

30, -10, 200, 0.5이 반환되었습니다.

|  |
| --- |
| =======Source Code======= |
| def calc(a, b):  return a+b, a-b, a\*b, a/b a = input("첫 번째 정수를 입력하시오: ") b = input("두 번째 정수를 입력하시오: ") add, sub, mul, div = calc(int(a), int(b)) print(f"{add}, {sub}, {mul}, {div}이 반환되었습니다.") |
| ========실행결과========= |
|  |

06. 사용자로부터 2개의 정수를 받아서 수학 문제를 만들어서 화면에 출력하는 함수를 작성하고 테스트하시오.

Ex) 실행결과

첫 번째 정수를 입력하시오: 10

두 번째 정수를 입력하시오: 20

정수 10+20의 합은?

|  |
| --- |
| =======Source Code======= |
| def add(a, b, result):  if a+b == result:  return "맞았습니다."  return "틀렸습니다." def sub(a, b, result):  if a-b == result:  return "맞았습니다."  return "틀렸습니다." def mul(a, b, result):  if a\*b == result:  return "맞았습니다."  return "틀렸습니다." def div(a, b, result):  if str(a/b) == result:  return "맞았습니다."  return "틀렸습니다." a = input("첫 번째 정수를 입력하시오: ") b = input("두 번째 정수를 입력하시오: ") print(f"정수 {a} + {b}의 합은?", end=" ") print(add(int(a), int(b), int(input()))) print(f"정수 {a} - {b}의 뺌은?", end=" ") print(sub(int(a), int(b), int(input()))) print(f"정수 {a} \* {b}의 곱은?", end=" ") print(mul(int(a), int(b), int(input()))) print(f"정수 {a} / {b}의 몫은?", end=" ") print(div(int(a), int(b), input())) |
| ========실행결과========= |
|  |

10. 주어진 정수가 소수인지를 검사하는 함수 testPrime(n)를 작성하고 이 함수를 호출하여서 2부터 100 사이의 소수를 출력하여 보자.

Ex)실행 결과

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

|  |
| --- |
| =======Source Code======= |
| def testPrime(n):  for i in range(2, int(n\*\*0.5)+1):  if n % i == 0:  return 0  return 1 for i in range(2, 101):  if testPrime(i):  print(i, end=" ") |
| ========실행결과========= |
|  |

13. 인간과 컴퓨터가 주사위 게임을 하는 프로그램을 작성해보자. 실행 결과는 다음과 같다.

Ex) 실행결과

=========== dice\_game() 호출 ===========

인간: 주사위값= 6

컴퓨터: 주사위값= 2

=========== dice\_game() 복귀 ===========

중단할까요? y/Ny

주사위는 자동으로 던져진다. 주사위 게임을 하나의 함수로 작성해보자.

|  |
| --- |
| =======Source Code======= |
| import random  def dice\_game():  print("=========== dice\_game() 호출 ===========")  person = random.randint(1, 6)  computer = random.randint(1, 6)  print(f"인간: 주사위값= {person}")  print(f"컴퓨터: 주사위값= {computer}")  print("=========== dice\_game() 복귀 ===========",end="\n\n")  exit = input("중단할까요? y/N")  if exit == 'y' or exit == 'Y':  return  else:  dice\_game()  return dice\_game() |
| ========실행결과========= |
|  |

15. 다음과 같이 터틀 그래픽을 이용하여 사각형을 안에서 바깥으로 확장하면서 그리는 프로그램을 작성하라.

건물, 건축자재이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| =======Source Code======= |
| import turtle t = turtle.Turtle() turtle.setup(width=500, height=500) t.shape("turtle")  def maze\_squre(size):  for \_ in range(40):  t.forward(size)  t.left(90)  size+=5  maze\_squre(0) input() |
| ========실행결과========= |
|  |

**과제 7 - 6장 자료구조 (리스트) 연습문제**

03. 리스트에 저장된 값으로 다음과 같이 별표를 출력하여 막대 그래프를 그리는 프로그램을 작성하라.

Ex) 실행결과

20 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1 \*

12 \*\*\*\*\*\*\*\*\*\*\*\*

9 \*\*\*\*\*\*\*\*\*\*\*\*\*

18 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |
| --- |
| =======Source Code======= |
| list = [20, 1, 12, 9, 18] for i in list:  s = '%-5s' % i  print(s, "\*"\*i) |
| ========실행결과========= |
|  |

09. 다음과 같은 정수들이 리스트에 저장되어 있다. 이 리스트와 터틀 그래픽을 이용하여 다음과 같은 그림을 그리는 코드를 작성하라.

aList = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

텍스트, 실외, 하늘, 일이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| =======Source Code======= |
| import turtle t = turtle.Turtle() turtle.setup(width=500, height=500) aList = [i\*10 for i in range(1,13)]  for i in aList:  t.forward(i)  t.stamp()  t.backward(i)  t.right(360/len(aList)) t.backward(1) input() |
| ========실행결과========= |
|  |

12. 각 요소가 “#”이고 크기가 3x4x5인 3차원 배열을 생성하는 프로그램을 작성하고 테스트하라. 리스트 함축을 적극적으로 사용해보자.

Ex) 실행결과

[[[‘#’, ‘#’, “#’, ‘#’], [‘#’, ‘#’, “#’, ‘#’], [‘#’, ‘#’, “#’, ‘#’]], [[‘#’, ‘#’, “#’, ‘#’], [‘#’, ‘#’, “#’, ‘#’], [‘#’, ‘#’, “#’, ‘#’]]]

|  |
| --- |
| =======Source Code======= |
| list = [[['#']\*5]\*4]\*3 for i in list:  print(i)  list = [[['#']\*4]\*3]\*2 for i in list:  print(i) |
| ========실행결과========= |
| (3x4x5)    (2x3x4)     * 문제 오타인 것 같습니다. 문제에는 3x4x5인데 실행결과는 2x3x4 배열이라   일단 둘 다 출력했습니다 |

14. 지뢰찾기는 예전에 윈도우에 무조건 포함되어 있어서 상당히 많은 사람들이 즐겼던 프로그램이다. 윈도우 7에서도 제어판에서 추가할 수 있다. 2차원의 게임판 안에 지뢰가 숨겨져 있고 이 지뢰를 모두 찾아내는 게임이다. 지뢰가 없는 곳을 클릭했을 때 숫자가 나오면 주변칸에 지뢰가 숨겨져 있다는 것을 의미한다. 예를 들어서 숫자가 2이면 주변칸에 지뢰가 두개 있다는 의미가 된다. 지뢰찾기 게임을 위한 기초 작업을 하여 보자. 10 x 10 크기의 2차원 리스트를 만들고 여기에 지뢰를 숨긴다. 지뢰가 아닌 곳은 .으로 표시하고 지뢰인 곳은 #으로 표시하여 보자. 어떤 칸이 지뢰일 확률은 난수를 발생시켜서 결정한다. 전체의 30%를 지뢰로 하고 싶으면 발생된 난수가 0.3보다 적은 경우에 현재 칸에 지뢰를 놓으면 된다.

Ex) 실행결과

|  |
| --- |
| =======Source Code======= |
| import random  mine = [['.']\*10 for \_ in range(10)] for i in range(10):  for j in range(10):  if random.random() < 0.3:  mine[i][j] = "#" for i in mine:  print(i) |
| ========실행결과========= |
|  |

16. 소수를 구하는데 고대의 그리스 수학자 에라스토스테네스에 의하여 개발된 에라스토스테네스의 체(Sieve of Erastosthenes)라는 알고리즘이 있다. 이 알고리즘은 정해진 범위안의 소수를 찾아주는 비교적 간단한 방법이다. 우리는 2부터 100사이의 소수를 찾는다고 하자. 이 알고리즘은 공백 리스트를 생성하여서 2부터 100까지의 숫자를 추가한다. 알고리즘은 가장 작은 소수인 2부터 시작한다. 상식적으로 2의 배수는 소수가 아닌 것이 확실하다. 정해진 범위 안에서 2의 배수(4, 6, 8, …)를 모두 찾아서 리스트에서 삭제하낟. 다시 3의 배수(6, 9, …)를 모두 찾아서 리스트에서 삭제한다. (100-1)의 배수까지 삭제한 후에도 리스트에 남아 있는 값은 소수가 된다. 이 알고리즘을 이용하여 2부터 100사이의 소수를 찾아보라.

Ex) 실행결과

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 None

|  |
| --- |
| =======Source Code======= |
| list = [i for i in range(2,101)] for i in list:  value = i\*2  while value < 101:  if value in list:  list.remove(value)  value+=i for i in list:  print(i, end=" ") print("None") |
| ========실행결과========= |
|  |