|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7차시 | 1 | 번 |  | 연습: ☑ | 과제 : □ | 평가 : □ |

|  |
| --- |
| 문제1) 다음 중 반복문에 대한 설명 중 틀린 것은?  (1) 코드의 재사용성을 높여줍니다.  **(2) 모든 예외 상황을 자동으로 처리합니다.**  (3) 코드의 가독성을 높여줍니다.  (4) 반복 작업을 자동화하여 코드의 효율성을 증가시킵니다.    문제2) 알고리즘을 사람이 사용하는 언어로 기술한 것을 무엇이라고 하는가?  (1) 컴파일러 (Compiler)  (2) 기계어 (Machine Language)  **(3) 의사코드 (Pseudo Code)**  (4) 소스 코드 (Source Code)    문제3) "버스가 정류장에 도착할 때까지 계속 기다린다"를 while문을 사용해 의사코드로 표현하시오.  **정답3)**  **while (버스가 정류장에 없으면):**  **짧은\_시간동안\_기다리기**  문제4) 다음 중 while 문의 조건식 사용에 대해 잘못된 것은 무엇입니까?  (1) 조건식은 참일 때 반복이 계속됩니다.  (2) 조건식이 항상 참이면 무한 루프가 발생할 수 있습니다.  **(3) 조건식은 반드시 정수형이어야 합니다.**  (4) 조건식은 부울 표현식이어야 합니다.    문제5) 다음 코드를 실행하면, 화면에 무엇이 출력되는지 설명하시오.  n=1  while "":  print(n)  n+=1  **정답5) 아무것도 출력 안됨. while문에서 빈 문자열은 False로 인지됨. 따라서 반복문이 실행 안됨**  문제6) 다음 for 반복은 몇 회 실행되는지 계산한다.  s = "hello\tworld"  for ch in s:  print(ch)  **정답6) 11번**  문제7) 다음 range() 함수의 결과를 적어본다.  (1) range(-5, -2)  **(1) for i in range(-5, -2):**  **print(i)**  **-5, -4, -3**  (2) range(7, 1, -3)  **(2) for i in range(7, 1, -3):**  **print(i)**  **7, 4** |

|  |
| --- |
| 문제8) 다음 for 반복은 몇 회 실행되는지 계산한다  for i in range(3):  print("hello")  **정답8) 3회**  문제9) 다음의 경우 for, while 반복문 중 어느 것이 보다 자연스러운 선택인가?  for, while 어느 반복을 선택하더라도 모두 가능하지만, 문법구조를 볼 때 보다 자연스러운 것을 선택하세요.  (1) 정해진 횟수를 반복할 경우  **(1) for 문**  (2) 조건식에 따라 반복할 경우  **(2) while 문**  문제10) 다음과 같은 for, while 반복문을 서로 중첩할 수 있다. OX로 답하시오.  for 변수1 in 순서가\_있는\_객체1:  while 조건식1:  코드\_블록  **정답10) O**  문제11) 다음 코드의 출력을 적으시오.  count=0  while count < 10:  print(count)  count += 1  if count==5:  break  **정답11)**  **0**  **1**  **2**  **3**  **4**  문제12) 사용자로부터 숫자 5개를 입력받고, 가장 큰 값을 찾아서 반환하는 함수를 작성하고, 가장 큰 입력값을 출력하는 프로그램을 작성하시오.  <요구사항>  - 입력되는 숫자 5개는 모두 0보다 큰 양수로 가정함.  - 자료의 정렬 알고리즘이 적용된 함수를 사용하지 않음.  **정답12)**  **print("5개 정수를 압력하면 가장 큰 값을 화면에 출력합니다.")**  **maxValue = 0**  **count = 1**  **while count < 6:**  **n = int(input("0보다 큰 정수 한개를 입력하세요: "))**  **if n > maxValue:**  **maxValue = n**  **count += 1**  **print("최대값 = ", maxValue)** |

|  |
| --- |
| 문제13. while 문을 사용해서 프로그램을 작성하시오.  (1) 100 ~ 199까지의 합계를 계산  **(1) sum = 0**  **n = 100**  **while n < 200:**  **sum += n**  **n += 1**  **print(sum)**  (2) 100 ~ 199까지 짝수의 합계를 계산  **(2) sum = 0**  **n = 100**  **while n < 200:**  **sum += n**  **n += 2**  **print(sum)**  (3) 100 ~ 199까지 3의 약수만 합계를 계산  **(3) sum = 0**  **n = 100**  **while n < 200:**  **if n % 3 == 0:**  **sum += n**  **n += 1**  **print(sum)**  문제14) 위 문제를 하나의 함수 getSum()을 작성해서 해결해 보세요. 코드의 중복없이 하나의 함수를 공통으로 사용하기 위해서 필요한 입력과 출력을 설계하세요.  **정답14)**  **start는 어떤 숫자부터 더하기 시작할 것인지 지정**  **end는 end - 1까지 처리**  **increase는 한 번씩 반복문이 실행될 때마다 증가시킬 n값**    **def getSum(start, end, increase):**  **sum = 0**  **n = start**  **while n < end:**  **sum += n**  **n += increase**  **return sum**  **sum = getSum(100, 200, 1) # 100~199까지 더함**  **print(sum)**  **getSum(100, 200, 2) # 100, 102, 104, ..., 198까지 더함**  **getSum(102, 200, 3) # 102, 105, 108, ..., 198까지 더함** |

|  |
| --- |
| 문제15) 사용자로부터 1~100사이의 정수를 입력받아 화면에 출력합니다.  - 1~100 범위 밖의 정수가 입력되면 다시 받기 (1 과 100 포함)  - 입력되는 숫자는 음수와 양수를 포함하는 정수로 제한함  **정답15)**  **n = int(input("1~100 사이의 정수를 한 개 입력하세요: "))**    **while n < 1 or n > 100:**  **n = int(input("1~100 사이의 정수를 한 개 입력하세요: "))**  **print(n)**    문제16) 정수의 약수를 화면에 출력하는 프로그램 작성.  - 정수를 한 개 인자로 전달받고, 약수를 화면에 모두 출력하는 함수 작성.  - 테스트로 12와 16을 실행해 볼 것.  **정답16)**  **def printDivisors(n):**  **for i in range(1, n + 1):**  **if n % i == 0:**  **print(i)**  **print("12의 약수들")**  **printDivisors(12)**  **print("\n16의 약수들")**  **printDivisors(16)** |

|  |
| --- |
| 문제17) 주사위에서 나올 수 있는 범위인 1~6사이의 정수 5개를 무작위로 출력하는 코드를 작성하세요.중복되는 숫자가 있는지 세어보세요. ( Hint : import random )  **정답17)**  **import random**  **count1 = 0 # 1이 나온 횟수**  **count2 = 0 # 2가 나온 횟수**  **count3 = 0 # 3이 나온 횟수**  **count4 = 0 # 4가 나온 횟수**  **count5 = 0 # 5가 나온 횟수**  **count6 = 0 # 6이 나온 횟수**  **for i in range(5):**  **n = random.randint(1, 6)**  **if n == 1:**  **count1 += 1**  **elif n == 2:**  **count2 += 1**  **elif n == 3:**  **count3 += 1**  **elif n == 4:**  **count4 += 1**  **elif n == 5:**  **count5 += 1**  **elif n == 6:**  **count6 += 1**  **# 반복되는 횟수를 출력**  **print(count1, count2, count3, count4, count5, count6)**  문제18) 정수 n1과 n2가 인자로 전달되면 n1, n1 + 1, n1 + 2, ..., n2 까지 각 정수의 약수들을 화면에 출력하는 함수를 구현한다. 이 함수를 이용해서 10~16까지의 약수 들을 출력해본다.  **정답18)**  **def printDivisors(n1, n2):**  **for n in range(n1, n2 + 1):**  **print(f"n = {n}")**  **for i in range(1, n + 1): # 약수 출력**  **if n % i == 0:**  **print(i, end = ' ')**    **print()printDivisors(10, 16)** |

|  |
| --- |
| 문제19) randint()함수를 이용해서 1~6까지의 무작위 숫자를 10개 생성하되, 4이하의 숫자만 화면에 출력한다. 무작위로 정수를 생성한 후에 4보다 큰 숫자가 나온 경우에는, 다시 무작위로 숫자를 생성하도록 한다.  **정답19)**  **import random**  **for i in range(10):**  **n = random.randint(1, 6)**  **while n > 4:**  **n = random.randint(1, 6)**  **print(n)**  문제20) 문자열을 매개 변수로 전달 받고, 마침표, 느낌표, 줄바꿈 기호를 제거하는 함수를 작성한다. 작성된 함수에 "...What a beautiful day!\n" 문자열을 전달하고 결과값을 받아 화면에 출력하는 프로그램을 구현한다.  **정답20)**  **def removeSomeSpecialChars(s):**  **news = "" # 반환할 문자열**  **for ch in s:**  **if ch != '.' and ch != '!' and ch != '\n':**  **news += ch**  **return news**  **s = removeSomeSpecialChars("...What a beautiful day!\n")**  **print(s)** |

|  |
| --- |
| 문제21) 주어진 정수 n의 모든 약수의 합을 구하는 SumOfDivisors(n) 함수를 구현한다. 이 함수를 사용해서 12,30을 전달해 약수의 합을 출력하는 프로그램을 작성한다.  **정답21)**  **def SumOfDivisors(n):**  **sum = 0**  **for i in range(1, n + 1):**  **if n % i == 0: # 약수**  **sum += i**  **return sum**  **print(f"12의 약수의 합 = {SumOfDivisors(12)}")**  **print(f"30의 약수의 합 = {SumOfDivisors(30)}")**  문제22) 이 100을 넘는 최초 값을 반환하는 함수를 구현하고, 이를 사용하는 프로그 램을 작성한다.    **정답22)**  **def getCube(n):**  **a = 1**  **while a \*\* 3 < n:**  **a += 1**  **return a**  **print(getCube(100))**    문제23) 1000~9999까지의 정수 중 한 개를 n이라고 가정한다. n의 천 단위 수를 n1, 백 단위 수를 n2, 십 단위 수를 n3, 일 단위 수를 n4라고 부르자.  4825에서 n1은 4, n2는 8, n3는 2, n4는 5가 된다. 1000~9999까지의 정수 중 에서 인 n을 모두 찾아서 출력하는 프로그램을 작성한다.  예를 들어, 8208 = 84 + 24 + 04 + 84 = 4096 + 16 + 0 + 4096 이므로 한 가지 답이 된다.  **정답23)**  **for i in range(1000, 10000):**  **n1 = i // 1000**  **n2 = (i - n1 \* 1000) // 100**  **n3 = (i - n1 \* 1000 - n2 \* 100) // 10**  **n4 = (i - n1 \* 1000 - n2 \* 100 + n3 \* 10)**  **if i == n1 \*\* 4 + n2 \*\* 4 + n3 \*\* 4 + n4 \*\* 4:**  **print(i)** |

|  |
| --- |
| 문제24) 정수 n을 입력 받고 n이 소수(prime number)인지 아닌지 확인하는 함수를 구현한다. 사용자로부터 정수 한 개를 입력 받고 이 함수를 이용해서 소수인지 아닌지 화면에 출력하는 프로그램을 작성한다.  **정답24)**  **def isPrimeNumber(n):**  **for i in range(2, n):**  **if n % i == 0: # 약수**  **return False**  **return True**  **n = int(input("소수인지 확인하고 싶은 정수 한 개를 입력하세요: "))**  **if isPrimeNumber(n):**  **print(f"{n}은 소수입니다")**  **else:**  **print(f"{n}은 소수가 아닙니다.")**  문제25)    **정답25)**  **for i in range(1, 10):**  **for j in range(10):**  **if i \* 10 + j + j \* 10 + i == 110:**  **print(f"n1: {i}, n2 = {j}")**  문제26) 동전을 던져서 앞/뒷면이 나오는 횟수를 세고, ½ 확률에 수렴하는지 확인하는 프로그램을 작성한다. 컴퓨터에서 동전을 던질 수는 없으므로, random.randint()  함수를 이용해서 두 개 숫자 중 한 개를 무작위로 생성하여 동전의 앞/뒷면을 대신한다. 100, 1000, 10000회 던져서 앞/뒷면이 나오는 횟수를 각각 출력한다.  <요구사항>  - 정해진 횟수만큼 동전을 던지고 앞/뒷면이 나오는 횟수를 출력하는 함수를 구현  - 동전을 던지는 횟수는 함수에 입력으로 전달  - 앞/뒷면이 나오는 확률을 구해서 각각 출력  **정답26)**  **import random**  **def flipCoin(num):**  **countFront = 0 # 앞면이 나오는 횟수**  **countBack = 0 # 뒷면이 나오는 횟수**  **for i in range(num): # num번 반복**  **if random.randint(0, 1) == 0:**  **countFront += 1**  **else:**  **countBack += 1**  **print(num, "번 동전을 던짐")**  **print("앞면이 나올 확률:", countFront / num)**  **print("뒷면이 나올 확률:", countBack / num)**  **flipCoin(100)**  **flipCoin(1000)**  **flipCoin(10000)** |

|  |
| --- |
| 문제27) 다음 표를 출력하는 프로그램을 작성하시오.    - 반복문을 사용하고, 같은 행에서 셀(cell)은 탭 문자로 분리  **정답27)**  **for i in range(1, 9):**  **if i < 8:**  **print(f"{i} \* n", end = '\t')**  **else:**  **print(f"{i} \* n")**    **for n in range(1, 11):**  **for i in range(1, 9):**  **if i < 8:**  **print(i \* n, end = '\t')**  **else:**  **print(i \* n)**  문제28) 다음의 복리 계산 방식을 사용하여 아래와 같이 출력하는 프로그램을 작성하시오.  출력 형식 -> 1년: 복리총액 1,024,265원, 총이자액 24,265원  - 입력데이터: 원금, 연이자율, 투자기간(년)  **정답28)**  **def compound\_interest(principal, annual\_rate, years):**  **# 연 이자율을 월 이자율로 변환**  **annual\_rate= annual\_rate / 100**  **# 총 이자를 초기화**  **total\_interest = 0**  **for year in range(1, years + 1):**  **#  해당 연도의 원금 계산**  **principal = principal \* (1 + annual\_rate)**  **# 연도별 이자 계산**  **interest = principal - (principal / (1 + annual\_rate))**  **# 총 이자 누적**  **print(f"{year}년: 복리총액 {int(principal):,}원, 총이자액 {int(interest):,}원")**    **principal = float(input("원금을 입력하세요: "))**  **annual\_rate = float(input("연 이자율을 입력하세요(%): "))**  **years = int(input("투자 기간(년)을 입력하세요: "))**  **compound\_interest(principal, annual\_rate, years)** |

|  |
| --- |
| 문제29) 1부터 45사이(45 포함)의 숫자 중 6개를 고르는 로또 번호 생성기를 프로그램 해보세요.  **정답29)**  **import random**  **def generate\_lotto\_numbers():**  **# 1부터 45까지의 숫자 중에서 중복 없이 6개 선택**  **lotto\_numbers = random.sample(range(1, 46), 6)**  **# 선택된 숫자를 오름차순으로 정렬**  **lotto\_numbers.sort()**  **return lotto\_numbers**    **# 로또 번호 생성 및 출력**  **lotto = generate\_lotto\_numbers()**  **print("로또 번호: ", lotto)** |