Рівень А

№ 1

Завдання:

Напишіть програму, яка обчислює індекс маси тіла (ІМТ) для особи. Спочатку додаток має зчитувати зріст та масу користувача. Для обчислення ІМТ використовується одна з 2 формул. Якщо зріст вимірюється в дюймах, а маса – у фунтах, ІМТ визначається так:

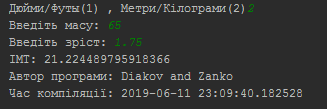
Для метрів та кілограмів ІМТ виражається відношеням

Виведіть відповідне значення ІМТ; вони будуть аналізуватись у наступних задачах.

Код:

imt\_method = str(input("Дюйми/Футы(1) , Метри/Кілограми(2)"))  
  
if imt\_method == "1":  
 weight = float(input("Введіть масу: "))  
 height = float(input("Введіть зріст: "))  
 imt1 = (703 \* weight) / (height \*\* 2)  
 print("IMT: " + str(imt1) )  
  
if imt\_method == "2":  
 weight = float(input("Введіть масу: "))  
 height = float(input("Введіть зріст: "))  
 imt2 = weight / (height \*\* 2)  
 print("IMT: " + str(imt2) )  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 2

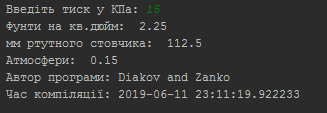
Завдання:

Створіть програму, яка зчитує тиск у кПа (кілопаскалях). Після зчитування даних, програма повинна повідомити про еквівалентний тиск у фт/дм2 (фунтах на квадратний дюйм), міліметрах ртутного стовпчика та атмосферах.

Код:

pressure\_kpa = float(input("Введіть тиск у КПа: "))  
psi = pressure\_kpa \* 0.15  
mm = pressure\_kpa \* 7.5  
atmo = pressure\_kpa \* 0.01  
print("Фунти на кв.дюйм: " , str(psi) )  
print("мм ртутного стовчика: " , str(mm))  
print("Атмосфери: " , str(atmo))  
  
import datetime   
def printTimeStamp(name):   
 print('Автор програми: ' + name)   
 print('Час компіляції: ' + str(datetime.datetime.now()))   
printTimeStamp('Diakov and Zanko')

Вивід:



№ 3

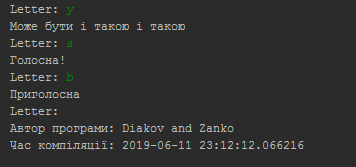
Завдання:

Створіть програму, яка зчитує букву англійського алфавіту від користувача. Якщо користувач ввів a, e, i, o або u, програма повинна вивести повідомлення, що введена буква – голосна. Якщо користувач вводить y, програма має повідомити, що інколи y – голосна, а інколи - приголосна. Інакше слід вивести повідомлення, що буква приголосна.

Код:

while True:  
 letter = str(input("Letter: "))  
  
 if letter == "y":  
 print("Може бути і такою і такою")  
 continue  
  
 elif letter == "a" and "e" and "i" and "o" and "u":  
 print("Голосна!")  
 continue  
  
 elif letter == "":  
 break  
  
 else:  
 print("Приголосна")  
 continue  
  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 4

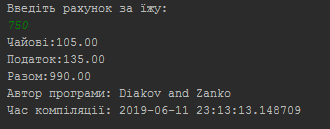
Завдання:

Програма в цій вправі повинна починати роботу зі зчитування вартості їжі, замовленої з ресторану (назва страви, кількість порцій, ціна за порцію). Потім програма обчислить податок та чайові за їжу. Розраховуйте на чайові в розмірі 14% від замовлення та податок у розмірі 18% від замовлення. Вивід вашої програми має включати розмір податку, чайових та загальну суму для оплати. Формат виводу передбачає представлення всіх значень з використанням 2 знаків після коми.

Код:

print("Введіть рахунок за їжу:")  
a = float(input())  
x = a \* 0.14  
y = a \* 0.18  
z = x + y + a  
print("Чайові:{:.2f}".format(x))  
print("Податок:{:.2f}".format(y))  
print("Разом:{:.2f}".format(z))  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 5

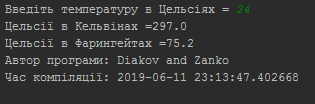
Завдання:

Напишіть програму, яка спочатку зчитує температуру в градусах Цельсія. Далі вона має вивести еквіваленту температуру в градусах Кельвіна та Фаренгейта.

Код:

temperature = float(input("Введіть температуру в Цельсіях = "))  
  
temp\_Kelvin =(temperature + 273)  
print("Цельсії в Кельвінах =" + str(temp\_Kelvin))  
temp\_Farengate =(temperature\*1.8 + 32)  
print("Цельсії в Фарингейтах =" + str(temp\_Farengate))  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 6

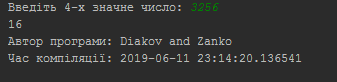
Завдання:

Напишіть програму, яка зчитуватиме 4-значне ціле число та буде виводити суму його цифр. Наприклад, якщо ввести 3141, програма має вивести 9 (=3+1+4+1).

Код:

num = input('Введіть 4-х значне число: ')  
  
num\_list = list(num)  
  
num\_1 = int(num\_list[0])  
num\_2 = int(num\_list[1])  
num\_3 = int(num\_list[2])  
num\_4 = int(num\_list[3])  
  
print(num\_1 + num\_2 + num\_3 + num\_4)  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 7

Завдання:

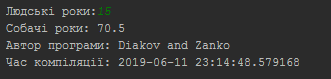
Поширеною думкою є те, що 1 рік людського життя еквівалентний 7 рокам життя собами. Проте це просте правило перестає виконуватись, коли собака стає дорослою (приблизно в 2 роки). У результаті деякі люди вважають, що краще вважати перші 2 роки життя собаки еквівалентними 10.5 рокам життя людини, а наступні – 4м рокам людини.

Напишіть програму, яка реалізує переведення з людських років у собачі за даними правилами. Ваш додаток має виводити попередження, коли вводиться від’ємне число.

Код:

import datetime  
  
chel = float(input("Людськi роки:"))  
if 0 < chel <= 2:  
 pes = 10.5 / chel  
 print("Собачi роки:", pes)  
elif chel > 2:  
 pes = 10.5 + (chel \* 4)  
 print("Собачi роки:", pes)  
elif chel < 0:  
 print("Введено вiд'ємне число")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 8

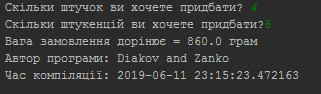
Завдання:

Онлайн-ритейлер продає 2 товари: штучки та штукенції. Кожна штучка важить 75г. Кожна штукенція важить 112г. Напишіть програму, яка зчитує кількість штучок та кількість штукенцій в замовленні від користувача. Потім програма має обчислювати та виводити на екран загальну масу замовлення.

Код:

import datetime  
  
shtuchku = float(input("Скільки штучок ви хочете придбати? "))  
shtukencii = float(input("Скільки штукенцій ви хочете придбати?"))  
  
mass =(shtuchku \* 75)+(shtukencii \* 112)  
print("Вага замовлення дорінює = " + str(mass),"грам")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 9

Завдання:

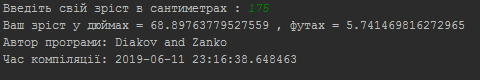
Багато людей думають про свій зріст у футах і дюймах, але у нас використовується метрична система. Напишіть програму, яка зчитує зріст у сантиметрах, а виводить у футах і дюймах.

Підказка: 1 фут = 12 дюймів. 1 дюйм = 2.54 см.

Код:

lenght = float(input("Введіть свій зріст в сантиметрах :"))  
  
duim = (lenght/2.54)  
futt = (duim/12)  
  
print("Ваш зріст у дюймах = " + str(duim), ", футах = " + str(futt))  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 10

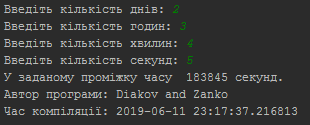
Завдання:

Напишіть програму, яка зчитує проміжок часу у вигляді кількості днів, годин, хвилин та секунд. Обчисліть та виведіть загальну кількість секунд у цьому проміжку часу.

Код:

days = int(input("Введіть кількість днів: "))  
hours = int(input("Введіть кількість годин: "))  
minutes = int(input("Введіть кількість хвилин: "))  
seconds = int(input("Введіть кількість секунд: "))  
all\_secs = (days \* 86400) + (hours \* 3600) + (minutes \* 60) + seconds  
print("У заданому проміжку часу " , str(all\_secs) + (" секунд."))  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 11

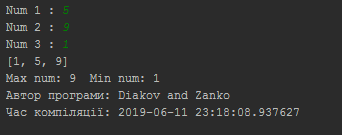
Завдання:

Напишіть програму, яка зчитує 3 цілих числа і сортує їх у порядку від меншого до більшого. Використовуйте функції min() та max() для знаходження найменшого та набільшого значень. Значення між ними можна обчислити, знайшовши суму всіх 3 чисел та віднявши мінімум та максимум.

Код:

num\_1 = int(input("Num 1 : "))  
num\_2 = int(input("Num 2 : "))  
num\_3 = int(input("Num 3 : "))  
num\_list = []  
num\_list.append(num\_1)  
num\_list.append(num\_2)  
num\_list.append(num\_3)  
num\_list.sort()  
maxi = max(num\_list)  
mini = min(num\_list)  
print(num\_list)  
print("Max num: " + str(maxi) , " Min num: " + str(mini))  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 12

Завдання:

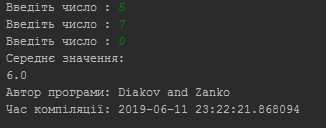
Створіть програму, яка обчислює середнє для набору значень, введеного користувачем. Користувач буде вводити 0 як індикатор того, що більше значень вводитись не буде. Ваша програма має виводити відповідне повідомлення про помилку, якщо перше введене значення буде нулем. Ця вправа не передбачає використання непримітивних типів даних.

Підказка: оскільки 0 позначає кінець вводу, він має не включатись в обчислення середньго.

Код:

tries = 0  
full = 0   
while True:  
 a = float(input("Введіть число : "))  
 if a == 0:  
 print("Середнє значення:")  
 print(full/tries)  
 break  
 else:  
 full += a  
 tries += 1  
 continue  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 13

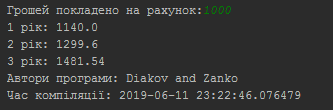
Завдання:

Уявімо, що Ви тільки-но відкрили новий накопичувальний рахунок зі ставкою 14% річних. Отриманий прибуток виплачується в кінці року та додається до балансу рахунку. Напишіть програму, що починає роботу зі зчитування кількості покладених на рахунок грошей. Далі вона має обчислити та вивести на екран суму на рахунку через 1, 2 і 3 роки. Показ сум обмежується 2 знаками після крапки.

Код:

import datetime  
  
start = float(input("Грошей покладено на рахунок:"))  
y1 = start + (start \* 0.14)  
y2 = y1 + (y1 \* 0.14)  
y3 = y2 + (y2 \* 0.14)  
print("1 рiк:", float('{:.2f}'.format(y1)))  
print("2 рiк:", float('{:.2f}'.format(y2)))  
print("3 рiк:", float('{:.2f}'.format(y3)))  
  
def printTimeStamp(name):  
 print('Автори програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 14

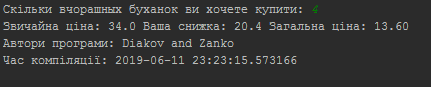
Завдання:

Пекарня продає буханки хлібу за 8.50 грн. На вчорашній хліб встановлено знижку на 60%. Напишіть програму, яка починає зі зчитування кількості буханок вчорашнього хлібу, які хоче придбати користувач. Потім програма має вивести на екран звичайну вартість товару, скидку за те, що він вчорашній, а також загальну суму покупки. Всі значення мають виводитись двома десятковими знаками після крапки та вирівнюватись по правому краю.

Код:

import datetime  
  
a = int(input("Скільки вчорашных буханок ви хочете купити: "))  
b = a \* 8.5  
c = (a \* 8.5)\*0.4  
d = b - c  
print("Звичайна ціна: " + str(b) + " Ваша снижка: " + str(d) + " Загальна ціна: " + str('{:.2f}'.format(c)))  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 15

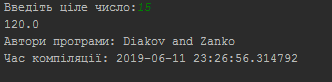
Завдання:

Напишіть програму, яка зчитує ціле число n від користувача та показує суму всіх цілих чисел від 1 до n. Формула для знаходження цієї суми:

Код:

import datetime  
  
n = int(input("Введiть цiле число:"))  
nast = n + 1  
mnoj = n \* nast  
dil = mnoj / 2  
print(dil)  
  
def printTimeStamp(name):  
 print('Автори програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 16

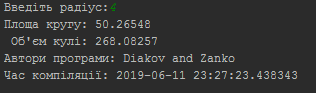
Завдання:

Напишіть програму, яка починає зі зчитування радіусу r від користувача. Програма продовжуватиме роботу, обислюючи та виводячи на екран площу кругу з радіусом r та об’єм кулі з радіусом r. Використовуйте константу pi з модуля math.

Код:

import datetime  
import math  
  
r = int(input("Введiть радiус:"))  
krug = math.pi \* (r\*\*2)  
kulya = 4/3 \* math.pi \* (r\*\*3)  
print("Площа кругу:", float('{:.5f}'.format(krug)), "\n", "Об'єм кулi:", float('{:.5f}'.format(kulya)))  
  
def printTimeStamp(name):  
 print('Автори програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 17

Завдання:

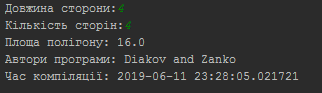
Полігон називають регулярним, якщо всі його сторони мають однакові довжини, а кути між усіма суміжними сторонами теж однакові. Площу регулярного полігону можна обчислити за формулою

де s – довжина сторони, а n – кількість сторін. Напишіть програму, яка зчитує ці значення і виводить на екран площу сконструйованого за даними значеннями полігону.

Код:

import datetime  
import math  
  
s = int(input("Довжина сторони:"))  
n = int(input("Кiлькiсть сторiн:"))  
pl = ((n \* (s\*\*2)) / (4 \* math.tan(math.pi / n)))  
print("Площа полiгону:", float('{:.5f}'.format(pl)))  
  
def printTimeStamp(name):  
 print('Автори програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 18

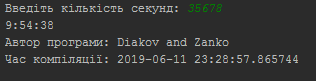
Завдання:

Напишіть програму, яка починає роботу зі зчитування кількості секунд від користувача. Далі вона має вивести еквівалентний проміжок часу у формі Д:ГГ:ХХ:СС (дні, години, хвилини, секунди). Потрібно, щоб форматований вивід передбачав по 2 знакових місця для годин, хвилин та секунд з дописуванням незначущого нуля за потреби.

Код:

a = int(input ("Введіть кількість секунд: "))  
h = str(a//3600)  
  
m = (a//60)%60  
s = a%60  
if m<10:  
 m='0'+str(m)  
else:  
 m=str(m)  
if s<10:  
 s='0'+str(s)  
else:  
 s=str(s)  
  
print(h +':' + m + ':' + s)  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 19

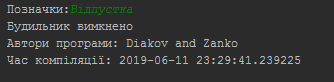
Завдання:

Припустимо, розумна програма може не вмикати будильник, щоб людина виспалась на вихідних чи у відпустці. Розробіть додаток, який буде приймати на вхід від користувача позначки про вихідний та відпустку, а виводитиме рішення, чи вмикати будильник. Вмикати його не слід, якщо день вихідний або людина у відпустці.

Код:

import datetime  
  
a = str(input("Позначки:"))  
if a == "Вiдпустка" or "вiдпустка":  
 print("Будильник вимкнено")  
elif a == "Вихiдний" or "вихiдний":  
 print("Будильник вимкнено")  
else:  
 print("Будильник ввiмкнено")  
  
  
def printTimeStamp(name):  
 print('Автори програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



Рівень В

№ 1

Завдання:

Закон ідеального газу – математичне наближення поведінки газів при зміні тиску, об’єму та температури. Зазвичай він описується рівнянням

де P – тиск у паскалях (Па), V – об’єм у літрах, n – молярна маса речовини, R = 8.314 mol K J – константа ідеального газу, T – температура в градусах Кельвіна.

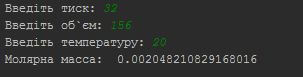
Напишіть програму, яка обчислює молярну масу газу, коли користувач задає тиск, об’єм і температуру. Протестуйте свою програму на визначення кількості молей газу в SCUBA tank. Типовий SCUBA tank містить 12 літрів газу під тиском 20,000,000 Па. Беремо кімнатну температуру (20º).

Підказка: Температура конвертується з Цельсія в Кельвіни шляхом додавання 273.15 градусів.

Код:

import datetime  
  
def printTimeStamp(name):  
  
 print('Автор програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
 printTimeStamp("Diakov and Zanko")  
  
p = int(input("Введіть тиск: "))  
v = int(input("Введіть об`єм: "))  
t = int(input("Введіть температуру: "))  
r = 8.314  
k = t + 273.15  
  
print("Молярна масса: ", (p\*(v/1000))/(r\*k))

Вивід:



№ 2

Завдання:

Нижче наведена таблиця містить діапазони магнітуд землетрусів за шкалою Ріхтера та опис відповідного землетрусу:

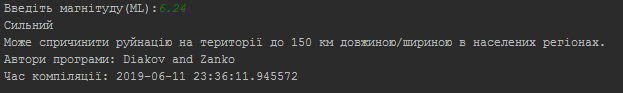
|  |  |
| --- | --- |
| Магнітуда | Опис |
| <2.0 | Мікро (micro) |
| 2.0-3.0 | Дуже слабкий (very minor) |
| 3.0-4.0 | Слабкий (minor) |
| 4.0-5.0 | Легкий (light) |
| 5.0-6.0 | Помірний (moderate) |
| 6.0-7.0 | Сильний (strong) |
| 7.0-8.0 | Дуже сильний (major) |
| 8.0-10.0 | Великий (great) |
| >=10.0 | Рідкісно великий (meteoric) |

Напишіть програму, яка зчитує від користувача магнітуду та виводить відповідний опис.

Код:

import datetime  
  
  
path = float(input('Введіть магнітуду(ML):'))  
  
if path <= 0 :  
 print('Землетрусу немаэ')  
elif path <= 2 :  
 print('Мікро \nМікроземлетруси, не відчуваються.')  
elif path >= 2.0 and path <= 3.0 :  
 print('Дуже слабкий \nЯк правило не відчуваються, але реєструються.')  
elif path >= 3.0 and path <= 4.0 :  
 print('Слабкий \nЧасто відчуваються, дуже рідко завдають шкоди.')  
elif path >= 4.0 and path <= 5.0 :  
 print('Легкий\nВідчутне тремтіння речей всередині будинків, значна шкода малоймовірна.')  
elif path >= 5.0 and path <= 6.0 :  
 print('Помірний \nМоже завдати значної шкоди старим та погано сконструйованим будівлям на незначній території.')  
elif path >= 6.0 and path <= 7.0 :  
 print('Сильний \nМоже спричинити руйнацію на території до 150 км довжиною/шириною в населених регіонах.')  
elif path >= 7.0 and path <= 8.0 :  
 print('Дуже сильний \nЗначна руйнація на значній території.')  
elif path >= 8.0 and path <= 10.0 :  
 print('Великий \nСерйозна руйнація на територіях довжиною/шириною в сотні кілометрів.')  
elif path >= 10.0 :  
 print('Рідкісно великий')  
  
  
  
def printTimeStamp(name):  
 print('Автори програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 3

Завдання:

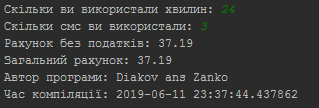
Деякий телефонний тариф включає 200 хвилин в мережі та 50 текстових повідомлень за ₴35.00 на місяць. Кожна додаткова хвилина коштує ₴0.17, а додаткова SMS - ₴0.15. Всі рахунки включають ₴0.44 внеску в Пенсійний фонд та 5%-вий податок.

Напишіть програму, яка зчитує кількість хвилин та текстових повідомлень, використаних за місяць. Виведіть базову плату за користування (без внесків та податків), а також загальний рахунок для користувача. Забезпечте виведення грошових сум з 2 знаками після коми.

Код:

xv = int(input("Скільки ви використали хвилин: "))  
sms = int(input("Скільки смс ви використали: "))  
a = 35  
b = (a\*0.05) + 0.44 + a  
  
if xv>200:  
 a += (xv-200)\*0.17  
  
if sms>50:  
 a += (sms - 50)\*0.17  
  
print("Рахунок без податків: " + str('{:.2f}'.format(b)))  
print("Загальний рахунок: " + str('{:.2f}'.format(b)))  
  
import datetime  
  
def printTimeStamp(name):  
  
 print('Автор програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov ans Zanko")

Вивід:



№ 4

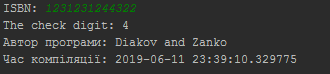
Завдання:

|  |  |
| --- | --- |
| 13-цифровий міжнародний стандартний номер книги (International Standard Book Number) – це унікальний код, який ідентифікує книгу, що продається. Остання цифра називається перевірочною цифрою (check digit) і використовується для перевірки на помилки. Для обчислення цього значення   * 1. кожну з 12 перших цифр почережно множать на 1 або 3,   2. додають добутки та ділять результат за модулем 10. Перевірочна цифра отримується відніманням від 10 отриманого результату.   Наприклад, для ISBN 9789609319614    Напишіть програму, що зчитуватиме 13-цифровий ISBN та перевірятиме останню цифру на валідність цього номеру. | 2,5 бала |

Код:

import datetime  
  
ibs = input("ISBN: ")  
numb = list(ibs)  
s = int(numb[0]) + int(numb[1])\*3 + int(numb[2]) + int(numb[3])\*3 + int(numb[4]) + int(numb[5])\*3 + int(numb[6]) + int(numb[7])\*3 + int(numb[8]) + int(numb[9])\*3 + int(numb[10]) + int(numb[11])\*3  
n = 10 -(s % 10)  
print("The check digit:", n)  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 5

Завдання:

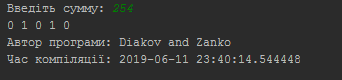
Розгляньте програмне забезпечення, яке працює на апараті для приготування кави. Одна із задач, яку воно повинно виконувати – визначення того, скільки здачі повертати з готівкової оплати товару.

Напишіть програму, яка починає роботу зі зчитування кількості копійок (ціле число). Далі програма має обчислювати та виводити на екран номінали монет та їх кількість для того, щоб видати здачу покупцю. Здача має віддаватись якомога меншою кількістю монет. Припускаємо, що машина завантажена монетами номіналами 25, 50 копійок, 1, 2 та 5 грн.

Код:

from math import floor  
  
  
def main():  
 while True:  
 a = int(input("Введіть сумму: "))  
 b = floor(a)  
  
 if b > 0:  
 break  
  
   
 g5 = b // 500  
 g2 = (b % 500) // 200  
 g1 = ((b % 500) % 200) // 100  
 k50 = (((b % 500) % 200) % 100) // 50  
 k25 = ((((b % 500) % 200) % 100) % 50) // 25  
  
 print(g5,g2,g1,k50,k25)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()  
  
import datetime  
  
def printTimeStamp(name):  
  
 print('Автор програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 6

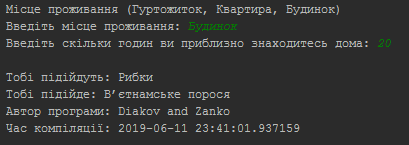
Завдання:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Напишіть програму, яка рекомендуватиме людині домашнього улюбленця відповідно до стилю життя його майбутнього власника.   |  |  |  | | --- | --- | --- | | **Місце проживання** | **Час вдома** | **Рекомендація** | | Будинок | Понад 18 годин | В’єтнамське порося | | Будинок | Від 10 до 17 годин | Собака | | Будинок | Менше 10 годин | Змія | | Квартира | Більше 10 годин | Кішка | | Квартира | Менше 10 годин | Хом’як | | Гуртожиток | Більше 6 годин | Рибки | | Гуртожиток | Менше 6 годин | Мурашник | |

Код:

print("Місце проживання (Гуртожиток, Квартира, Будинок)")  
home = input('Введіть місце проживання: ')  
  
time = int(input('Введіть скільки годин ви приблизно знаходитесь дома: '))  
print(' ')  
  
if home =='Гуртожиток':  
   
 if time < 6:  
 print('Тобі підійде: Мурашник')  
   
elif time == 6 :  
 print('Тобі підійде: Мурашник або рибки')  
   
if time > 6:  
 print('Тобі підійдуть: Рибки')  
  
if home =='Квартира':  
   
 if time <10 :  
 print("Тобі підійде: Хом'як")  
  
  
  
if home =='Квартира' and time == 10 :  
 print("Тобі підійде: Хом'як або Кішка")  
  
  
if home =='Квартира' and time > 10:  
 print('Тобі підійде: Кішка')  
  
  
if home =='Будинок' and time <=10 :  
 print("Тобі підійде: Змія")   
  
  
if home =='Будинок' and time >10 and time <17:  
 print("Тобі підійде: Собака")  
  
  
if home =='Будинок' and time > 18:  
 print('Тобі підійде: В’єтнамське порося')  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 8

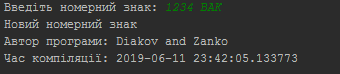
Завдання:

У деяких штатах США старіші номерні знаки складаються з 3 великих літер, за якими йдуть 3 цифри. Коли всі можливі номерні знаки були вичерпані, перейшли до формату «4 цифри + 3 великі букви». Напишіть програму, яка зчитує рядок символів, а потім визначає, чи підходить введений номер під якийсь із форматів. Якщо це не так, слід повідомити користувача про помилку.

Код:

import datetime  
  
path = input("Введіть номерний знак: ")  
bukwy = path[0:3]  
cifry = path[3:7]  
cifry2 = path[0:4]  
bukwy2 = path[4:7]  
old\_bukwy = bukwy.isupper()  
old\_cifry = cifry.isdigit()  
new\_cifry = cifry2.isdigit()  
new\_bukwy = bukwy2.isupper()  
  
if old\_bukwy == True and old\_cifry == True:  
 print("Старий номерний знак")  
elif new\_cifry == True and new\_bukwy == True:  
 print("Новий номерний знак")  
else:  
 print("Номер неправильний")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 9

Завдання:

Напишіть програму, яка конвертує десяткове число в двійкове. Зчитуйте десяткове число від користувача як ціле число, а потім використовуйте алгоритм ділення для виконання перетворення.

**Ініціалізувати змінну result порожнім рядком**

**Оголосити змінну q – число для перетворення**

**repeat**

**Оголосити r та присвоїти йому значення остачі від ділення q на 2**

**Звести r до рядкового типу та додати її до початку**

**result**

**Поділити націло q на 2 та зберегти результат у q**

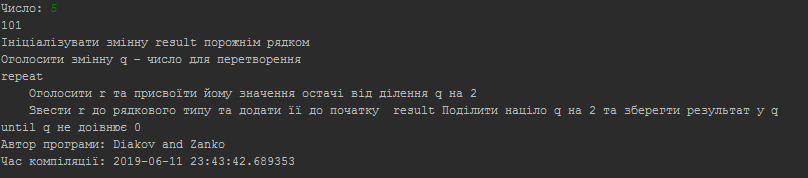
**until q не доівнює 0**

Коли алгоритм завершує роботу, змінна result міститиме двійкове представлення числа. Відобразіть результат на екрані з відповідним пояснювальним текстом.

Код:

result = ''  
q = int(input("Число: "))  
while q > 0:  
 r = q % 2  
 result = str(r) + result  
 q = q // 2  
print(result, "\nІніціалізувати змінну result порожнім рядком\nОголосити змінну q – число для перетворення\nrepeat\n Оголосити r та присвоїти йому значення остачі від ділення q на 2\n Звести r до рядкового типу та додати її до початку result Поділити націло q на 2 та зберегти результат у q\nuntil q не доівнює 0")  
  
import datetime  
  
def printTimeStamp(name):  
  
 print('Автор програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 10

Завдання:

Поверхня Землі викривлена, і відстань між градусами довготи варіюється залежно від широти. У результаті, знаходження відстані між 2 точками на поверхні Землі дещо ускладнюється.

Нехай (t1, g1) та (t2, g2) – широти і довготи 2 точок на поверхні планети. Відстань між ними в кілометрах визначається за формулою:

**Відстань = 6371.01 × arccos(sin(t1) × sin(t2) + cos(t1) × cos(t2) × cos(g1 − g2))**

Величина 6371.01 – середній радіус планети Земля в кілометрах.

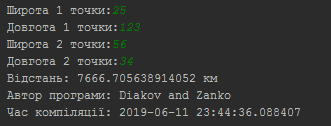
Напишіть додаток, що дозволятиме користувачу вводити широту і довготу 2 точок у градусах. Ваша програма має виводити відстань між точками в кілометрах.

Підказка: тригонометричні функції в Python працюють з радіанною мірою. Як результат, буде потрібно конвертувати користувацький ввід з градусів у радіани до обчислення відстані. Модуль math містить функцію radians(), яка переводить градуси в радіани.

Код:

import math  
import datetime  
  
shir = math.radians(float(input("Широта 1 точки:")))  
dovg = math.radians(float(input("Довгота 1 точки:")))  
shir2 = math.radians(float(input("Широта 2 точки:")))  
dovg2 = math.radians(float(input("Довгота 2 точки:")))  
vidst = 6371.01 \* math.acos(math.sin(shir) \* math.sin(shir2) + math.cos(shir) \* math.cos(shir2) \* math.cos(dovg - dovg2))  
print("Вiдстань:", vidst, "км")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 11

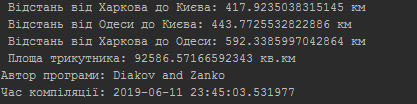
Завдання:

На основі попередньої задачі 6 визначте площу трикутника, утвореного містами Львів, Одеса та Харків. Пропонується визначити відстань між цими містами, а потім встановити площу за формулою Герона.

Код:

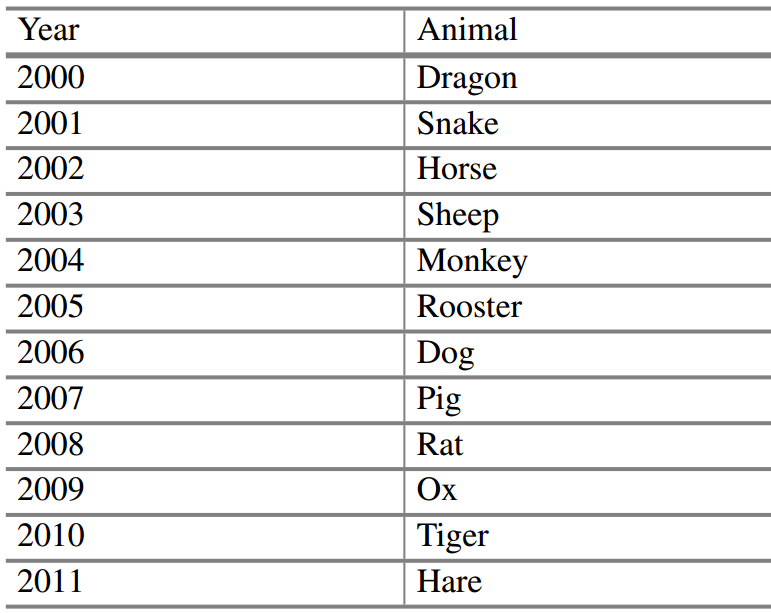
import math  
import datetime  
  
shir = math.radians(50)  
dovg = math.radians(36.15)  
shir2 = math.radians(50.27)  
dovg2 = math.radians(30.30)  
shir3 = math.radians(46.28)  
dovg3 = math.radians(30.43)  
  
vidst1 = 6371.01 \* math.acos(math.sin(shir) \* math.sin(shir2) + math.cos(shir) \* math.cos(shir2) \* math.cos(dovg - dovg2))  
vidst2 = 6371.01 \* math.acos(math.sin(shir2) \* math.sin(shir3) + math.cos(shir2) \* math.cos(shir3) \* math.cos(dovg2 - dovg3))  
vidst3 = 6371.01 \* math.acos(math.sin(shir) \* math.sin(shir3) + math.cos(shir) \* math.cos(shir3) \* math.cos(dovg - dovg3))  
  
p = (vidst1 + vidst2 + vidst3) / 2  
s = math.sqrt(p \* (p - vidst1) \* (p - vidst2) \* (p - vidst3))  
  
print(" Вiдстань вiд Харкова до Києва:", vidst1, "км", "\n", "Вiдстань вiд Одеси до Києва:", vidst2, "км", "\n", "Вiдстань вiд Харкова до Одеси:", vidst3, "км", "\n", "Площа трикутника:", s, "кв.км")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№12

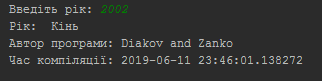
Китайський гороскоп ставить у відповідність 12-річному циклу назви тварин:



Напишіть програму, яка зчитує рік і виводить тварину, що йому відповідає. Програма має коректно працювати з роками нашої ери, а не лише з представленими в таблиці.

Код:

x = ["Дракон","Змія","Кінь","Вівця","Мавпа","Птах","Пес","Свиня","Миша","Бик","Тигр","Заєць"]  
y = int(input("Введіть рік: "))  
if y >=2000 and y <= 2011:  
 print("Рік: ", x[y-2000])  
elif y >2011:  
 while y >= 2011:  
 y-=12  
 print("Рік: "+x[y-2000])  
elif y <2000:  
 while y <= 2000:  
 y+=12  
 print("Рік: "+x[y-2000])  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")



№ 14

Завдання:

Коли дмухає вітер в холодну погоду, повітря відчувається ще холоднішим, ніж є насправді. Це відбувається тому, що рух повітря пришвидшує охолодження теплих об’єктів, таких як люди. Даний ефект називають прохолодністю вітру.

У 2001 році в Канаді, Великобританії та США було впроваджено формулу, яка обчислює індекс прохолодності вітру (wind chill index):

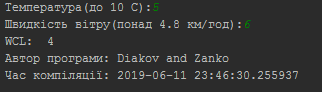
У даній формулі – температура повітря в градусах Цельсія, – швидкість вітру в км/год.

Напишіть програму, яка спочатку зчитує температуру повітря та швидкість вітру, а потім виводить індекс прохолодності вітру, округлений до найближчого цілого. Індекс прохолодності вітру вважається коректним за умови, якщо температура не перевищує 10ºС, а швидкість вітру буде понад 4.8 км/год. Дані умови потрібно врахувати при виводі інформації.

Код:

import datetime  
  
t = float(input("Температура(до 10 C):"))  
v = float(input("Швидкiсть вiтру(понад 4.8 км/год):"))  
if t <= 10 and v > 4.8:  
 index = 13.12 + (0.6215 \* t) - (11.37 \* (v\*\*0.16)) + (0.3965 \* t \* (v\*\*0.16))  
 print("WCL: ", round(index))  
else:  
 print("Некоректнi даннi")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 15

Завдання:

Високосні роки визначаються за наступними правилами:

• Будь-який рік, що націло ділиться на 400, високосний.

• З решти всі роки, що діляться на 100 – невисокосні.

• З решти всі роки, що діляться на 4, є високосними.

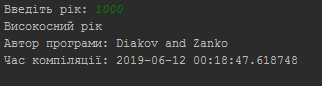
• Всі інші - невисокосні.

Напишіть програму, яка зчитує рік та виводить повідомлення про те, чи є цей рік високосним.

Код:

year = int(input("Введіть рік: "))  
if year % 400 == 0:  
 print("Звичайний рік")  
elif year % 100 == 0:  
 if year % 4 == 0:  
 print("Високосний рік")  
 else:  
 print("Звичайний рік")  
  
import datetime  
  
def printTimeStamp(name):  
  
 print('Автор програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 17

Завдання:

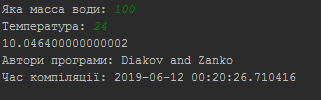
Кількість енергії, необхідної для збільшення температури 1 граму матеріалу на 1ºС, називається теплоємністю C. Загальна енергія, потрібна для підняття температури m грамів матеріалу на ΔT градусів Цельсія можна обчислити за формулою:

Напишіть програму, яка зчитує масу води та температурну зміну від користувача. Ваша програма має відображати загальну кількість енергії, яку потрібно додати чи забрати, щоб досягнути заданої зміни температури.

Код:

import datetime  
  
m = int(input("Яка масса води: "))  
T = int(input("Температура: "))  
C = 4.186  
q = (m/1000)\*T\*C  
print(q)  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 19

Завдання:

У деякому університеті використовується буквенна система оцінювання у відповідності до балів із таблиці:

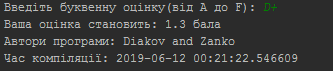
|  |  |
| --- | --- |
| **Буквенна оцінка** | **Бальна оцінка** |
| A+ | >4.0 |
| A | 4.0 |
| A- | 3.7 |
| B+ | 3.3 |
| B | 3.0 |
| B- | 2.7 |
| C+ | 2.3 |
| C | 2.0 |
| C- | 1.7 |
| D+ | 1.3 |
| D | 1.0 |
| F | 0 |

Напишіть програму, яка зчитуватиме буквенну оцінку від користувача. Далі вона виводить на екран бальну оцінку, що відповідає введеній буквенній. Забезпечте коректну роботу для ситуації, коли введена буквенна оцінка не передбачена в таблиці.

Код:

mark = input('Введіть буквенну оцінку(від A до F): ')  
  
if mark == 'A+' :  
 print('Оцінка більше 4 балів')  
elif mark == 'A' :  
 print('Ваша оцінка становить: 4.0 бала')  
elif mark == 'A-' :  
 print('Ваша оцінка становить: 3.7 бала')  
elif mark == 'B+' :  
 print('Ваша оцінка становить: 3.3 бала')  
elif mark == 'B' :  
 print('Ваша оцінка становить: 3.0 бала')  
elif mark == 'B-' :  
 print('Ваша оцінка становить: 2.7 бала')  
elif mark == 'C+' :  
 print('Ваша оцінка становить: 2.3 бала')  
elif mark == 'C' :  
 print('Ваша оцінка становить: 2.0 бала')  
elif mark == 'C-' :  
 print('Ваша оцінка становить: 1.7 бала')  
elif mark == 'D+' :  
 print('Ваша оцінка становить: 1.3 бала')  
elif mark == 'D' :  
 print('Ваша оцінка становить: 1.0 бал')  
elif mark == 'F' :  
 print('Ваша оцінка становить: 0 балiв')  
else:  
 print('Помилка, введіть буквенну оцінку від A до F')  
  
  
import datetime  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 20

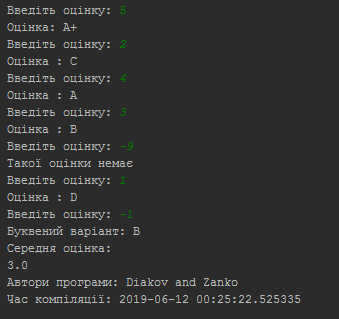
Завдання:

Попередня вправа включала таблицю, яка показувала правила конвертації буквенної оцінки в бали для деякої академічної установи. У даній вправі Вам потрібно написати програму, яка дозволятиме користувачеві вводити оцінки студентів групи, доки не буде введено -1. У процесі програма має конвертувати оцінку в її буквенний еквівалент на виводити на екран. Наприкінці програма має обчислити середню оцінку по групі, вивести її та її буквенний еквівалент.

Код:

a = 0  
b = 0  
while True:  
 mark = float(input("Введіть оцінку: "))  
 a += 1  
 b += mark  
 if mark > 4:  
 print("Оцінка: А+")  
 elif 3.7 < mark <= 4:  
 print("Оцінка : А")  
 elif 3.3 < mark <= 3.7:  
 print("Оцінка : A-")  
 elif 3.0 < mark <= 3.3:  
 print("Оцінка : B+")  
 elif 2.7 < mark <= 3:  
 print("Оцінка : B")  
 elif 2.3 < mark <= 2.7:  
 print("Оцінка : B-")  
 elif 2 < mark <= 2.3:  
 print("Оцінка : C+")  
 elif 1.7 < mark <= 2:  
 print("Оцінка : C")  
 elif 1.3 < mark <= 1.7:  
 print("Оцінка : C-")  
 elif 1 < mark <= 1.3:  
 print("Оцінка : D+")  
 elif 0 < mark <= 1:  
 print("Оцінка : D")  
 elif mark == 0:  
 print("Оцінка : F")  
 elif mark == -1:  
 a -= 1  
 b -= mark  
 break  
 else:  
 a -= 1  
 b -= mark  
 print("Такої оцінки немає")  
c =(b/a)\*10  
if c in range(0,9):  
 print("Буквений варіант: F")  
elif c in range(10, 12):  
 print("Буквений варіант: D")  
elif c in range(13, 16):  
 print("Буквений варіант: D+")  
elif c in range(17, 19):  
 print("Буквений варіант: C-")  
elif c in range(20, 22):  
 print("Буквений варіант: C")  
elif c in range(23, 26):  
 print("Буквений варіант: C+")  
elif c in range(27, 29):  
 print("Буквений варіант: B-")  
elif c in range(30, 32):  
 print("Буквений варіант: B")  
elif c in range(33, 36):  
 print("Буквений варіант: B+")  
elif c in range(37, 39):  
 print("Буквений варіант: A-")  
elif c == 40:  
 print("Буквений варіант: A")  
elif c > 40:  
 print("Буквений варіант: A+")  
  
print("Середня оцінка: ")  
d = c/10  
print(d)  
  
import datetime  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 21

Завдання:

Найбільший спілдьний дільник двох додатніх цілих чисел (n, m) є найбільшим числом d, на яке вони діляться без остачі. Один з алгоритмів для знаходження НСД:

**Ініціалізувати d значенням меншого з чисел m і n.**

**While на d націло не діляться m і n do**

**Зменшити значення d на 1**

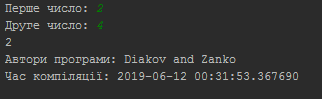
**Вивести d**

Напишіть програму, яка зчитує 2 цілих додатніх числа та знаходить найбільший спільний дільник за алгоритмом.

Код:

import datetime  
  
n = int(input("Перше число: "))  
m = int(input("Друге число: "))  
while n != 0 and m != 0:  
 if n > m:  
 n %= m  
 else:  
 m %= n  
print(n+m)  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

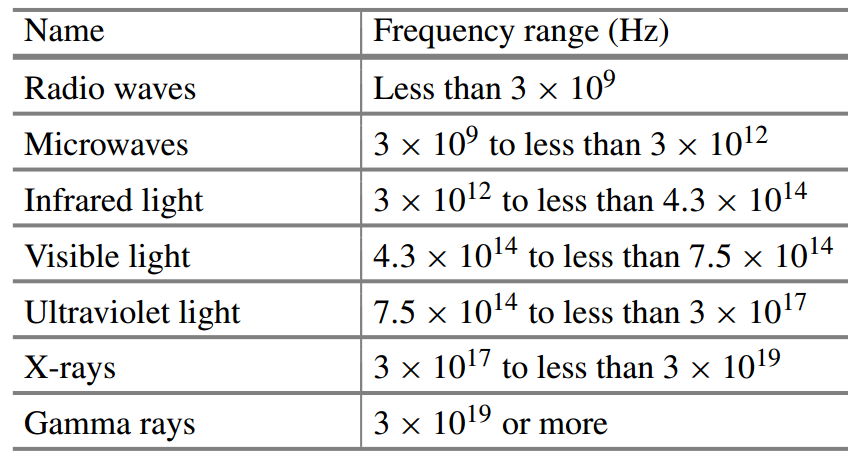
Вивід:



№ 22

Завдання:

Електромагнітне випромінювання можна розділити на 7 категорій за частотою:

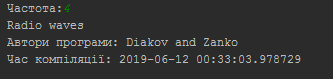


Напишіть програму, яка зчитує частоту радіації та виводить відповідну назву хвиль.

Код:

import datetime  
  
chast = int(input("Частота:"))  
  
  
if 0 < chast < 3 \* (10\*\*9):  
 print("Radio waves")  
elif 3 \* (10\*\*9) <= chast < 3 \* (10\*\*12):  
 print("Microwaves")  
elif 3 \*(10\*\*10) <= chast < 4.3 \* (10\*\*14):  
 print("Infrared light")  
elif 4.3 \* (10\*\*14) <= chast < 7.5 \* (10\*\*14):  
 print("Visible light")  
elif 7.5 \* (10\*\*14) <= chast < 3 \* (10\*\*17):  
 print("Ultraviolet light")  
elif 3 \* (10\*\*17) <= chast < 3 \* (10\*\*19):  
 print("X-rays")  
elif chast >= 3 \* (10\*\*19):  
 print("Gamma rays")  
else:  
 print("Частоту введено неправильно!")  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 23

Завдання:

Компанія, що займається постачанням води, обчислює клієнтські рахунки за наступною формулою:

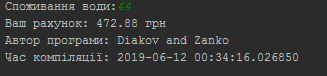
1. Фіксований внесок – 20 грн.
2. За перші 30 кубометрів плата складає 9.86 грн./м3
3. За наступні 20 кубометрів – 11.22 грн./м3
4. За наступні 10 кубометрів – 13.06 грн./м3
5. За додаткові кубометри – 17.89 грн./м3

Напишіть програму, яка буде зчитувати споживання води в кубометрах та виводити на екран рахунок.

Код:

import datetime  
voda = float(input("Споживання води:"))  
  
if voda <= 30:  
 a = 20 + (voda \* 9.86)  
 print("Ваш рахунок:", a, "грн")  
elif 30 < voda <= 50:  
 b = 20 + (30 \* 9.86) + ((voda - 30) \* 11.22)  
 print("Ваш рахунок:", b, "грн")  
elif 50 < voda <= 60:  
 c = 20 + (30 \* 9.86) + (20 \* 11.22) + ((voda - 50) \* 13.06)  
 print("Ваш рахунок:", c, "грн")  
elif voda > 60:  
 d = 20 + (30 \* 9.86) + (20 \* 11.22) + (10 \* 13.06) + ((voda - 60) \* 17.89)  
 print("Ваш рахунок:", d, "грн")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

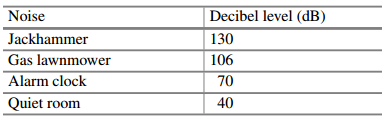
Вивід:



№ 24

Завдання:

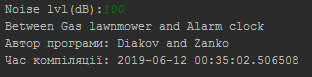
Наступна таблиця перелічує рівень гучності в децибелах для кількох поширених шумів. Напишіть програму, яка зчитує рівень шуму в децибелах від користувача. Якщо користувач вводить рівень шуму, що співпадає з табличним значенням, виведіть повідомлення, що містить лише тип шуму. Якщо користувач вводить кількість децибелів, яка знаходиться в проміжку між шумами, програма повідомлятиме, між якими шумами розташовується даний рівень шуму. Забезпечте коректну роботу програми при значеннях, нижчих за найтихіший шум, та значеннях, що перевищують найгучніший шум з таблиці.



Код:

import datetime  
  
db = int(input("Noise lvl(dB):"))  
if db > 130:  
 print("More than Jackhammer")  
elif db == 130:  
 print("Jackhammer")  
elif 106 < db < 130:  
 print("Between Jackhammer and Gas lawnmower")  
elif db == 106:  
 print("Gas lawnmower")  
elif 70 < db < 106:  
 print("Between Gas lawnmower and Alarm clock")  
elif db == 70:  
 print("Alarm clock")  
elif 40 < db < 70:  
 print("Between Alarm clock and Quiet room")  
elif db == 40:  
 print("Quiet room")  
elif db < 40:  
 print("Lower than Quiet room")  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 25

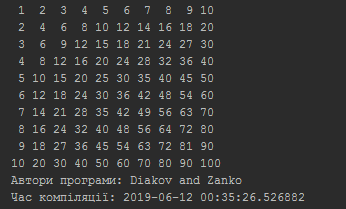
Завдання:

Напишіть програму, яка відобразить табличку множення 10х10, як показано на рисунку:

Код:

import datetime  
  
for i in range(1, 11):  
 for j in range(1, 11):  
 print("{:2d}".format(i \* j), end=" ")  
 print()  
  
def printTimeStamp(name):  
  
 print('Автори програми: ' + name)  
  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
  
printTimeStamp("Diakov and Zanko")

Вивід:



№ 26

Завдання:

Факторизація цілого числа n може виконуватись за алгоритмом:

**factor = 2**

**While factor <= n do**

**If n націло ділиться на factor then**

**значить factor є множником n**

**поділити n на factor націло,**

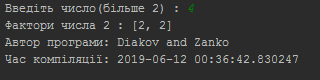
**Else**

**збільшити factor на 1**

Напишіть програму, яка зчитує ціле число . Якщо число менше 2, програма має повідомити про помилку. Інакше додаток обчислює прості числа, на які можна розкласти .

Код:

import datetime  
  
def factorise(n):  
 factors = []  
 factor = 2  
 if n < factor:  
 print("Помилка")  
 return 0  
 else:  
 while( factor \* factor <= n) and (n >= 2):  
 if n % factor == 0:  
 factors.append(factor)  
 n = n // factor  
 else:  
 factor = factor + 1  
  
 factors.append(n)  
 print("Фактори числа", n, ":", factors)  
   
n = int(input("Введiть число(бiльше 2) : "))  
  
factorise(n)  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:

№ 27

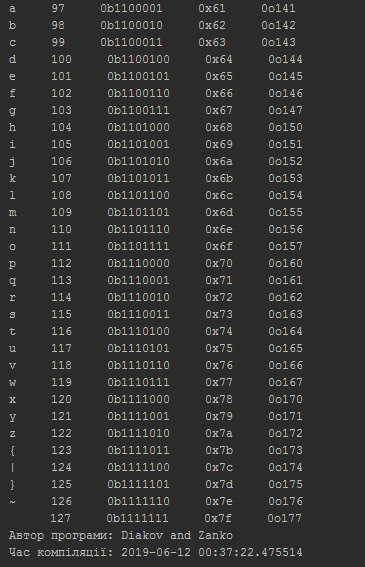
Завдання:

Виведіть на екран символи таблиці ASCII від 0 до 127 разом з їх номером у десятковій, двійковій, вісімковій та шістнадцятковій системах числення.

Код:

for i in range (0,128):  
 b = chr(i)  
 r = ord(b)  
 c = bin(i)  
 t = hex(i)  
 f = oct(i)  
 print( b, " ", r, " ", c, " ", t, " ", f)  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:



Рівень С

№ 1

Завдання:

Рулетка має 38 комірок: 18 чорних, 18 червоних і 2 зелених. Зелені комірки позначаються як 0 та 00. Червоними комірками є 1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30 32, 34 та 36. Решта цілих чисел з діапазону від 1 до 36 є чорними.

На рулетку можна зробити багато ставок. Для цієї вправи розглянемо наступні можливі ставки:

• Одне число (від 1 до 36, 0 або 00)

• Червоне vs Чорне

• Парне vs Непарне (0 та 00 не включаються)

• Від 1 до 18 vs від 19 до 36

Напишіть програму, яка симулює кручення колеса рулетки за допомогою генератора випадкових чисел. Виведіть вибране число та всі ставки, які потрібно оплатити. Наприклад, для 13:

На рулетці випало 13...

Виплатити 13

Виплатити Black

Виплатити Odd

Виплатити 1 to 18

Якщо результати симуляції випадають на 0 або 00, слід вивести Виплатити 0 або Виплатити 00 без іншої інформації.

Код:

import datetime, random  
allplases = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, "0", "00"]  
reds = [1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30, 32, 34, 36]  
blacks = [2, 4, 6, 8, 10, 11, 13, 15, 17, 20, 22, 24, 26, 28, 29, 31, 33, 35]  
  
def oneNumber(num):  
 win = random.choice(allplases)  
 if (win == "0" or win == "00") and num == win:  
 return win  
 else:  
 print("На рулетці випало", win)  
 if win == int(num):  
 print("Виплатити", win)  
 else:  
 print("Ви програли")  
  
def red\_black(color):  
 win = random.choice(allplases)  
 print("На рулетці випало", "Червоне" if win in reds else "Чорне" if win in blacks else win)  
 print("Виплатити Червоне") if ((win in reds) and color == "Червоне") else print("Виплатити Чорне") if (win in blacks) and color == "Чорне" else print("Ви програли")  
  
def parne\_neparne(choice):  
 win = random.randint(1, 36)  
 print("На рулетці випало", win)  
 if ((win % 2 == 0) and choice == "Парне"):  
 print("Виплатити Парне")  
 elif (win % 2 != 0) and choice == "Непарне":  
 print("Виплатити Непарне")  
 else:  
 print("Ви програли")  
  
def inHalf(choice):  
 win = random.choice(allplases)  
 print("На рулетці випало", win)  
 if (win <= 18 and choice == "First"):  
 print("Виплатити 1 to 18")  
 elif win >= 19 and choice == "Second":  
 print("Виплатити 19 to 36")  
 else:  
 print("Ви програли")  
  
while True:  
 betOneNumber = oneNumber(input("\n{}\nВведіть число для ставки 'Одне число' (від 1 до 36, 0 або 00): ".format("="\*15)))  
 if betOneNumber != None:  
 print("Виплатити", betOneNumber)  
 continue  
 betRed\_Black = red\_black(input("Введіть колір для ставки 'Червоне vs Чорне' (Червоне або Чорне): "))  
 betparne\_neparne = parne\_neparne(input("Введіть вибір для ставки 'Парне vs Непарне' (Парне або Непарне): "))  
 betInHalf = inHalf(input("Введіть вибір для ставки 'Від 1 до 18 vs від 19 до 36' (First або Second): "))  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp("Diakov and Zanko")

Вивід:

№ 2

Завдання:

Напишіть програму, яка зчитує дату від користувача та обчислює дату наступного дня. Дата буде вводитись у числовій формі трьома окремими виразами (рік, місяць, день). Наприклад, якщо користувач вводить значення, яке представляє день 2017-05-10, Ваша програма має показати повідомлення, яке говоритиме, що наступним днем буде 2017-05-11.

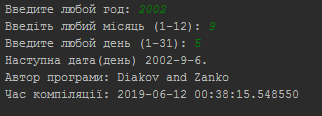
* Якщо користувач вводить значення на зразок 2017-04-30, програма має вказати, що наступним днем буде 2017-05-01.
* Якщо користувач вводить значення, що представляє дату 2016-12-31, програма має вивести наступний день - 2017-01-01.
* Забезпечте коректну роботу для високосних років.

Не використовуйте вбудованих методів для обчислення наступного дня!

Код:

year = int(input("Введите любой год: "))  
if (year % 400 == 0):  
 vus\_year = True  
elif (year % 100 == 0):  
 vus\_year = False  
elif (year % 4 == 0):  
 vus\_year = True  
else:  
 vus\_year = False  
  
month = int(input("Введіть любий місяць (1-12): "))  
if month in (1, 3, 5, 7, 8, 10, 12):  
 month\_long = 31  
elif month== 2:  
 if vus\_year:  
 month\_long = 29  
 else:  
 month\_long = 28  
else:  
 month\_long = 30  
day = int(input("Введите любой день (1-31): "))  
if day < month\_long:  
 day += 1  
else:  
 day = 1  
 if month == 12:  
 month = 1  
 year += 1  
 else:  
 month += 1  
print("Наступна дата(день) %d-%d-%d." % (year, month, day))  
for year in (year,month,day,):  
 if day > month\_long:  
 break  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:



№ 4

Завдання:

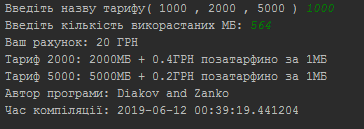
Інтернет-провайдер має три тарифи для населення:

|  |  |
| --- | --- |
| Тариф 1000 | 1000 Мб на місяць за 20 грн, позатарифно: 1Мб = 0.05 грн. |
| Тариф 2000 | 2000 Мб на місяць за 35 грн, позатарифно: 1Мб = 0.04 грн. |
| Тариф 5000 | 5000 Мб на місяць за 85 грн., позатарифно: 1 Мб = 0.02 грн. |

Напишіть програму, яка повинна отримувати цифру-тариф та кількість витрачених мегабайтів, а виводити підсумковий рахунок за місяць. Якщо використовується тариф 1000 або 2000, виведіть додатково інформацію про те, скільки б платили користувачі, якби перейшли на більш дорогі тарифи.

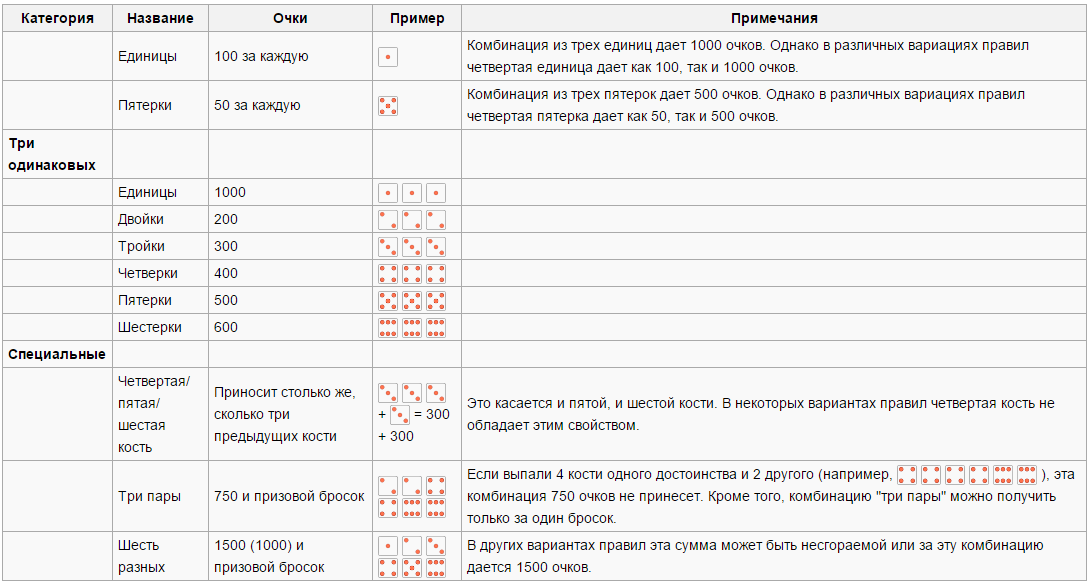
Код:

tar = input("Введіть назву тарифу( 1000 , 2000 , 5000 ) ")  
  
  
#1000mb  
if tar == "1000":  
 mb\_status = float(input("Введіть кількість викорастаних МБ: "))  
 if mb\_status < 0:  
 print("Input Error")  
 elif mb\_status <= 1000:  
 print("Ваш рахунок: 20 ГРН ")  
 elif mb\_status > 1000:  
 out\_tar = (mb\_status - 1000) \* 0.05  
 all\_cost = 20 + out\_tar  
 print("Ваш рахунок: " + str(all\_cost) + " ГРН" )  
 print("Тариф 2000: 2000МБ + 0.4ГРН позатарфино за 1МБ \nТариф 5000: 5000МБ + 0.2ГРН позатарфино за 1МБ")  
   
  
  
#2000mb  
if tar == "2000":  
 mb\_status = float(input("Введіть кількість викорастаних МБ: "))  
 if mb\_status < 0:  
 print("Input Error")  
 elif mb\_status <= 2000:  
 print("Ваш рахунок: 35 ГРН ")  
 elif mb\_status > 2000:  
 out\_tar = (mb\_status - 2000) \* 0.04  
 all\_cost = 35 + out\_tar  
 print("Ваш рахунок: " + str(all\_cost) + " ГРН" )  
 print("Тариф 2000: 2000МБ + 0.4ГРН позатарфино за 1МБ \nТариф 5000: 5000МБ + 0.2ГРН позатарфино за 1МБ")  
   
  
  
#5000mb  
if tar == "5000":  
 mb\_status = float(input("Введіть кількість викорастаних МБ: "))  
 if mb\_status < 0:  
 print("Input Error")  
 elif mb\_status <= 5000:  
 print("Ваш рахунок: 85 ГРН ")  
 elif mb\_status > 5000:  
 out\_tar = (mb\_status - 5000) \* 0.02  
 all\_cost = 85 + out\_tar  
 print("Ваш рахунок: " + str(all\_cost) + " ГРН" )  
  
  
import datetime  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

Вивід:

№5

Для гри в зонк потрібно 6 шестигранних кубиків, які будуть кидатись одночасно. Доступні наступні комбінації:



Напишіть підпрограму, яка аналізує кидок одного гравця. При цьому генерується 6 випадкових чисел (від 1 до 6). Виведіть на екран доступні комбінації за результатом кидка та очки, які за них набере користувач.

Код:

import random, datetime  
  
point = 0  
trys = 1  
while trys == 1:  
 trys -= 1  
 kubiki= [random.randint(1,6) for i in range(6)]  
 print(kubiki)  
 point += kubiki.count(1) \* 100  
 point += kubiki.count(5) \* 50  
 break\_group = 0  
 for i in range (1,7):  
 if kubiki.count(i) >=3:  
 print("{} однакові кості '{}' +{}".format(kubiki.count(i), i, (kubiki.count(i)- 2) \* 100 \*(10 if i == 1 else i)))  
 point += (kubiki.count(i)-2) \* 100 \*(10 if i == 1 else 1)  
 break\_group += 1  
 if len(set(kubiki)) == 3 and break\_group == 0:  
 print("Три пари, +750 points та ще одна спроба")  
 point += 750  
 trys += 1  
 if len(set(kubiki)) == 6:  
 print("Три пари, +1500 points та ще одна спроба")  
 point += 1500  
 trys +=1  
print("Ваші points", point)  
  
def printTimeStamp(name):  
 print('Автор програми: ' + name)  
 print('Час компіляції: ' + str(datetime.datetime.now()))  
printTimeStamp('Diakov and Zanko')

