**HAUB SCHOOL OF BUSINESS**

**SAINT JOSEPH’S UNIVERSITY**

**DSS 615: Python Programming**

**Instructor: Michael Ghen**

**Assignment 7**

By:

Vinayak Suresh Tayshetye (10673718)

1.

def main():

input = getInput()

resultA = "The maximum height of the ball is {:.2f} feet."

resultB = "The ball will hit the ground after approximately {:.2f} seconds."

print(resultA.format(maxHt(input[0], input[1])))

print(resultB.format(time(input[0], input[1])))

def getInput():

h = eval(input("Enter the initial height of the ball: "))

v = eval(input("Enter the initial velocity of the ball: "))

if (isValid(h, v)) == False:

print ("Please type a positive number.")

getInput()

return h, v

def isValid(x, y):

if (x > 0) and (y > 0):

result = True

else:

result = False

return result

def maxHt(h, v):

maxHeight = h + (v \* v/32) - (16 \* ((v/32) \*\* 2))

return maxHeight

def time(h, v):

t = 0.1

ht = h

while ht > 0:

t += .01

ht = h + (v \* t) - (16 \* (t \*\* 2))

return t

main()

2.

def main():

number = int(input("Enter a positive integer > 1: "))

while number <= 1:

num = int(input("Enter a positive integer > 1: "))

print("Largest prime factor: " + str(primeFac(number)[-1]))

print("Smallest prime factor: " + str(primeFac(number)[0]))

def primeFac(num):

f = 2

primeNumbers = []

while num > 1:

if num % f == 0:

num = num / f

primeNumbers.append(f)

else:

f += 1

return primes[0], primes[-1]

main()

3.

def main():

number = int(input("Enter a number with maximum 27 digits: "))

verbalizeNumber(number)

def verbalizeNumber(num):

if int(num) > 0 and len(str(num)) <= 27:

num1 = "0" \* (27 - len(str(num))) + str(num) #To make the digits of lenght 27 if it is less.

print("{:>4d} septillion".format(int(num1[:3])))

print("{:>4d} sextillion".format(int(num1[3:6])))

print("{:>4d} quintillion".format(int(num1[6:9])))

print("{:>4d} quadrillion".format(int(num1[9:12])))

print("{:>4d} trillion".format(int(num1[12:15])))

print("{:>4d} billion".format(int(num1[15:18])))

print("{:>4d} million".format(int(num1[18:21])))

print("{:>4d} thousand".format(int(num1[21:24])))

print("{:>4d}".format(int(num1[24:27])))

else:

print("Number is invalid")

main()

4.

#Q4

def main():

itemName = input("Enter name of item purchased: ")

years = eval(input("Enter year purchased: "))

cost = eval(input("Enter cost of item: "))

life = eval(input("Enter estimated life of item (in years): "))

method = input("Enter method of depreciation (SL or DDB): ")

description(itemName, years, cost, life, method)

depreciationMethod(itemName, years, cost, life, method)

def description(itemName, years, cost, life, method):

print("\nDescription: {:s}".format(itemName))

print("Year of purchase: {:d}".format(years))

print("Cost: ${:,.2f}".format(cost))

print("Estimated life: {:d} years".format(life))

if method.upper() == 'DDB':

print("Method of depreciation: double-declining balance")

else:

print("Method of depreciation: single-line balance")

def depreciationMethod(itemName, years, cost, life, method):

if method.upper() == 'SL':

methodVal = 1

else:

methodVal = 2

begCost = cost

depreciation = begCost \* (methodVal / life)

totalDeprecn = depreciation

print("\n{0:5s}{1:>15s}{2:>20s}{3:>23s}".format("", "Value at", "Amount Deprec", "Total Depreciation"))

print("{0:4s}{1:>17s}{2:>19s}{3:>22s}".format("", "Beg of Yr.", "During Year", "to End of Year"))

for year in range(life):

if year == (life - 1):

print("{0:4d}{1:>17,.2f}{2:>19,.2f}{3:>22,.2f}".format(years, cost, cost, cost))

else:

print("{0:4d}{1:>17,.2f}{2:>19,.2f}{3:>22,.2f}".format(years, begCost, depreciation, totalDeprecn))

begCost = begCost - (begCost \* (methodVal / life))

depreciation = begCost \* (methodVal / life)

totalDeprecn += depreciation

years += 1

main()

5.

def main():

word = str(input("Enter a word: "))

if isTripleConsecutive(word):

print("{:s} contains three successive letters \nin consecutive alphabetical order.".format(word))

else:

print("{:s} does not contain three successive letters \nin consecutive alphabetical order.".format(word))

def isTripleConsecutive(word):

word = word.upper()

for i in range( len(word) - 2):

if ord(word[i]) == (ord(word[i + 1]) - 1):

if ord(word[i + 1]) == (ord(word[i + 2]) - 1):

booleanVal = True

break

else:

booleanVal = False

else:

booleanVal = False

return booleanVal

main()

6.

#Q6

def main():

global ISBN

getISBN()

if (ISBN[:(len(ISBN)-1)].isdigit() and (ISBN[-1] == 'X') or ISBN[len(ISBN)-1].isdigit()):

if checkISBN(ISBN):

print("The number is valid.")

else:

print("The number is not valid.")

else:

main6()

def getISBN():

ISBN = input("Enter an ISBN: ")

ISBN = ISBN.replace('-', '')

#listISBN1 = list(ISBN)

length = len(ISBN)

if length == 10:

return ISBN

else:

print("Enter a 10 digit ISBN")

getISBN()

def checkISBN(ISBN):

listISBN1 = list(ISBN)

listISBN = ['10' if ch == 'X' else ch for ch in listISBN1]

total = 0

i = 10

while i >= 1:

for letter in listISBN:

total += int(letter) \* i

i -= 1

if total % 11 == 0:

result = True

else:

result = False

return result

main()