

Programming in Python (CSE 3142)

MINOR ASSIGNMENT-3: CONTROL STRUCTURES

-
1. Write an assignment statement using a single conditional expression for the following if-else code:

```
if marks >= 70:  
    remarks = 'good'  
else:  
    remarks = 'average'
```
 2. It is commonly said that one human year is equivalent to 7 dog years. However, this simple conversion must recognize that dogs reach adulthood in approximately two years. As a result, some people believe that it is better to count each of the first two human years as 10.5 dog years and then count each additional human year as 4 dog years. Write a program that implements the conversion from human to dog years described in the previous paragraph. Ensure that your program works correctly for conversions of less than two human years and conversions of two or more human years. Your program should display an appropriate error message if the user enters a negative number.
 3. The length of a month varies from 28 to 31 days. Create a program that reads the month's name from the user as a string. Then your program should display the number of days in that month. Display "28 or 29 days" for February so that leap years is addressed.
 4. A roulette wheel has 38 spaces on it. Of these spaces, 18 are black, 18 are red, and two are green. The green spaces are numbered 0 and 00. The red spaces are numbered 1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30 32, 34 and 36. The remaining integers between 1 and 36 are used to number the black spaces. Many different bets can be placed in roulette. We will only consider the following subset of them in this exercise:
 - Single number (1 to 36, 0, or 00)
 - Red versus Black
 - Odd versus Even (Note that 0 and 00 do not pay out for even)
 - 1 to 18 versus 19 to 36

Write a program that simulates a spin of a roulette wheel by using Python's random number generator. Display the number that was selected and all of the bets that must be payed. For example, if 13 is selected then your program should display:

The spin resulted in 13...

Pay 13

Pay Black

Pay Odd

Pay 1 to 18

If the simulation results in 0 or 00 then your program should display Pay 0 or Pay 00 without any further output

5. The following table lists the sound level in decibels for several common noises. Write a program that reads a sound level in decibels from the user. If the user enters a decibel level that matches one of the noises in the table then your program should display a message containing only that noise. If the user enters a number of decibels between the noises listed then your program should display a message indicating which noises the value is between. Ensure that your program also generates reasonable

output for a value smaller than the quietest noise in the table, and for a value larger than the loudest

Noise	Decibel Level
Jackhammer	130dB
Gas Lawnmower	106 dB
Alarm Clock	70 dB
Quiet Room	40 dB

noise in the table.

6. Write a python script to assign a grade to a student based on marks obtained as per the criteria mentioned in the above table:

Range	Grade
marks>=90 and <=100	A
marks>=70 and <=89	B
marks>=50 and <=69	C
marks>=40 and <=49	D
marks>=0 and <=39	F

7. Study the program segments given below. In each case, give the output produced, if any.

a. total = 0

count = 20

 while count > 5:

 total += count

 count -= 1

 print(total)

b. total = 0

N = 5

for i in range(1, N+1):

 for j in range(1, i+1):

 total += 1

 print(total)

c. total = 0

N = 10

for i in range(1, N+1):

 for j in range(1, i+1):

 total += 1

 print(total)

d. total = 0

N = 5

for i in range(1, N+1):

 for j in range(1, i+1):

 total += 1

 total -= 1

 print(total)

e. total = 0

N = 5

for i in range(1, N+1):

 for j in range(1, N+1):

```
total += i
print(total)
f. total = 0
N = 5
for i in range(1, N+1):
    for j in range(1, i+1):
        total += j
    print(total)
g. total = 0
N = 5
for i in range(1, N+1):
    for j in range(1, N+1):
        total += i+j
    print(total)
h. total = 0
N = 5
for i in range(1, N+1):
    for j in range(1, i+1):
        for k in range(1, j+1):
            total += 1
    print(total)
i. number = 72958476
a, b = 0, 0
while (number > 0):
    digit = number % 10
    if(digit % 2 != 0):
        a += digit
    else:
        b += digit
    number /= 10
print(a,b)
```

8. Write a function to determine whether a given natural number is perfect. A natural number is said to be a perfect number if it is the sum of its divisors. For Example, 6 is a perfect number because $6 = 1+2+3$, but 15 is not because $15 \neq 1+3+5$.
9. Write a program to find the maximum of three numbers using a nested function.
10. Write a function that takes two numbers as input parameters and returns their least common multiple.
11. Write a function that takes two numbers as input parameters and returns their greatest common divisor.
12. Write a function that accepts as an input parameter the number of rows to be printed and prints a figure like:

(a)		(b)
1 1 2 1 2 3 1 2 3 4 1 2 3 4 5		1 2 1 2 3 2 1 2 3 4 3 2 1 2 3 4
(c)		(d)
5 4 3 2 1 4 3 2 1 3 2 1 2 1 1		1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
(e)		(f)
1 2 3 4 5 2 3 4 5 3 4 5 4 5 5	*	*
(g)		(h)
*	*	*
*	*	*
*	*	*
*	*	*
*	*	*
(i)		(j)
*	*	*
*	*	*
*	*	*
*	*	*
*	*	*
(k)		(l)
*	*	*
*	*	*
*	*	*
*	*	*
*	*	*
(m)		(n)
5 5		5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

13. Write a function that finds the sum of the n terms of the following series:

a.

$$1 - x^2/2! + x^4/4! - x^6/6! + \dots x^n/n!$$

b.

$$e^x = 1 + x/1! + x^2/2! + x^3/3! + \dots$$

14. Write a function that returns True or False depending on whether the given number is a palindrome.

15. Write a function that returns the sum of digits of a number, passed to it as an argument.

16. Write a program that prints Armstrong numbers in the range 1 to 1000. An Armstrong number is a number whose sum of the cubes of the digits is equal to the number itself. For Example,

$$370 = 3^3 + 7^3 + 0^3$$

17. Write a function that takes two numbers as input parameters and returns True or False depending on whether they are co-primes. Two numbers are said to be co-prime if they do not have any common divisor other than one.

18. Write a function to multiply two non-negative numbers by repeated addition, for example, $7*5 = 7+7+7+7+7$.

19. Write a program that computes the average of a collection of values entered by the user. The user will enter 0 as a sentinel value to indicate that no further values will be provided. The program should display an appropriate error message if the first value entered by the user is 0

20. A particular retailer is having a 60 percent off sale on a variety of discontinued products. The retailer would like to help its customers determine the reduced price of the merchandise by having a printed discount table on the shelf that shows the original prices and the prices after the discount has been applied. Write a program that uses a loop to generate this table, showing the original price, the discount amount, and the new price for purchases of \$4.95, \$9.95, \$14.95, \$19.95 and \$24.95. Ensure that the discount amounts and the new prices are rounded to 2 decimal places when displayed.

21. Write a program that reads an integer from the user. If the value entered by the user is less than 2 then your program should display an appropriate error message. Otherwise, your program should display the prime numbers that can be multiplied together to compute n, with one factor appearing on each line. For example:

Enter an integer (2 or greater): 72

The prime factors of 72 are:

2

2

2

3

3

MINOR ASSIGNMENT -3

1. marks = int(input("Enter Marks:"))
print({True: "Good", False: "Average"} [marks >= 70])

Output

Enter marks : 85

Good

2. def years(n):
 if (n<0):
 print("Invalid Age.")
 elif (n>0 and n<=1):
 return n*7
 elif (n>1 and n<=2):
 return 10.5
 elif (n>2):
 return 10.5+(4*n)

print("Dog years:", years(int(input("Enter human years:"))))

Output

Enter human years : 15

Dog years : 70.5

3. def days():

month = input("Enter month name:")

if (month.lower() == "january" or month.lower() == "march"
or month.lower() == "may" or month.lower() == "july"
or month.lower() == "august" or month.lower() == "october"
or month.lower() == "december"):

 print("31 days")

elif (month.lower() == "february"):

 year = int(input("Enter year:"))

 if ((year % 400 == 0) or (year % 100 != 0)
 and (year % 4 == 0)):

 print("29 Days")

 else:

 print("28 Days")

 else:

 print("30 Days")

days()

Output:-

Enter month name: february

Enter year: 2004

29 Days.

4. import random

```
spin_result = random.randint(0, 38)
print(f'The spin resulted in {spin_result}...')

if spin_result == 0:
    print("Pay 0")
elif spin_result == 00:
    print("Pay 00")
else:
    print(f'Pay {spin_result}')

if spin_result in [1, 3, 5, 7, 9, 12, 14, 16, 18, 19, 21, 23,
                   25, 27, 30, 32, 34, 36]:
    print('Pay', Red, Black)
else:
    print('Pay', Black, Red)

if spin_result % 2 == 0 and spin_result != 0:
    print('Pay Even')
else:
    print('Pay Odd')

if spin_result >= 1 and spin_result <= 18:
    print('Pay 1 to 18')
else:
    print('Pay 19 to 36')
```

Output:-

The spin resulted in 14...

Pay 14
Pay Red
Pay Even
Pay 1 to 18

```
5. def noise():
    db = int(input("Enter sound level:"))
    if (db > 130):
        print("Noise = Loudest")
    elif (db == 130):
        print("Noise: Jackhammer")
    elif (db < 130 and db > 106):
        print("Noise: Between Jackhammer and gas
              lawnmover")
    elif (db == 106):
        print("Noise: Gas lawnmover")
    elif (db < 106 and db > 70):
        print("Noise: Between Gas lawnmover and
              alarm clock")
    elif (db == 70):
        print("Noise: Alarm clock")
    elif (db < 70 and db > 40):
        print("Noise: Between quiet Quiet Room and
              alarm clock")
    elif (db == 40):
        print("Noise: Quiet Room")
    elif (db < 40):
        print("Noise: quietest")
```

noise()

Output

Enter noise level: 120

Noise: Between Jackhammer and Gas Lawnmover.

6. def grade(marks):

```
    if (marks >= 90 and marks <= 100):  
        print ("Grade: A")
```

```
    elif (marks >= 70 and marks <= 89):  
        print ("Grade: B")
```

```
    elif (marks >= 50 and marks <= 69):  
        print ("Grade: C")
```

```
    elif (marks >= 40 and marks <= 49):  
        print ("Grade: D")
```

```
    elif (marks >= 0 and marks <= 39):  
        print ("Grade: F")
```

else:

```
    print ("Invalid Marks Entered")
```

mark = int(input("Enter Marks:"))

grade(mark)

Output

Enter Marks: 82

Grade: B

7. a) total = 0
count = 20
while count > 5
 total += count
 count -= 1
print (total)
- b) total = 0
N = 5
for i in range (1, N+1):
 for j in range (1, i+1):
 total += 1
print (total)
- c) total = 0
N = 10
for i in range (1, N+1):
 for j in range (1, i+1):
 total += 1
print (total)
- d) total = 0
N = 5
for i in range (1, N+1):
 for j in range (1, i+1):
 total += 1
 total -= 1
print (total)
- }
- Output
- No output: Infinite Loop
- Output
- 15
- Output
- 55
- Output
- 10

e) $\text{total} = 0$

$N = 5$

for i in range ($1, N+1$):

 for j in range ($1, N+1$):

$\text{total} += i$

 print (total)

Output

75

f) $\text{total} = 0$

$N = 5$

for i in range ($1, N+1$):

 for j in range ($1, i+1$):

$\text{total} += j$

 print (total)

Output

35

g) $\text{total} = 0$

$N = 5$

for i in range ($1, N+1$):

 for j in range ($1, N+1$):

$\text{total} += i+j$

 print (total)

Output:

150

h) $\text{total} = 0$

$N = 5$

for i in range ($1, N+1$):

 for j in range ($1, i+1$):

 for k in range ($1, j+1$):

$\text{total} += 1$

 print (total)

Output:

35

i) number = 72958476

a, b = 0, 0

while (number > 0):

output

digit = number % 10

47.3333... 6

if (digit % 2 == 0):

a += digit

else:

b += digit

number /= 10

print(a, b)

8. def isperfect(n):

sum = 1

for i in range(2, ~~n~~ int(n**0.5)+1):

if (n % i == 0):

sum = sum + i + n // i

if (sum == n):

print(n, "is a perfect number.")

else:

print(n, "is not a perfect number.")

~~n = int(input("Enter a number:"))~~

~~isperfect(n)~~

Output

Enter a number: 6

6 is a perfect number.

9. def man3(a, b, c):
 def man2(x, y):
 if x > y:
 return x
 else:
 return y
 return man2(man2(a, b), c)

a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = int(input("Enter c:"))
print("The maximum of three numbers is : ",
 man3(a, b, c))

Output

Enter a: 5

Enter b: 4

Enter c: 9

The maximum of three numbers is : 9

10. def lcm(x, y):
 if x > y:
 greater = x
 else:
 greater = y
 while (True):
 if ((greater % x == 0) and (greater % y == 0)):
 lcm = greater
 break

greater += 1
return lcm

num1 = int(input("Enter num1:"))

num2 = int(input("Enter num2:"))

print("The LCM is:", lcm(num1, num2))

11. def gcd(x, y):

if x > y:

smaller = y

else

smaller = x

for i in range(1, smaller + 1):

if ((x % i == 0) and (y % i == 0)):

gcd = i

return gcd.

num1 = ~~10~~ int(input("Enter num1:"))

num2 = ~~20~~ int(input("Enter num2:"))

print("The GCD is:", gcd(num1, num2))

Output

Enter num1: 10

Enter num2: 20

The G.C.D. is 10

Output

Enter num1: 15

Enter num2: 12

The LCM is 60

12.

a) def print_pattern(n):
for i in range(1, n+1):
 for j in range(1, i+1):
 print(j, end=' ')
 print()

print-pattern(int(input("Enter n: ")))

Output

Enter n: 5

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

b) def print_pattern(n):

for i in range(1, n+1):
 for j in range(1, n-i+1):
 print(" ", end=" ")
 for k in range(i, 0, -1):
 print(k, end=" ")
 for l in range(2, i+1):
 print(l, end=" ")
 print()

print-pattern(int(input("Enter n: ")))

Output

Enter n: 4

```
    1  
   2 2 2  
  3 2 1 2 3  
4 3 2 1 2 3 4
```

```
c> def print_pattern(n):  
    for i in range(0, n):  
        for j in range(n-i, 0, -1):  
            print(j, end=' ')  
        print()  
print(pattern(int(input("Enter n: "))))
```

Output

Enter n: 5

```
5 4 3 2 1  
4 3 2 1  
3 2 1  
2 1  
1
```

```
d> def print_pattern(n):  
    for i in range(1, n+1):  
        for j in range(1, i+1):  
            print(i, end=' ')  
        print()  
print(pattern(int(input("Enter n: "))))
```

Output

Enter n : 5

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

e) def print_pattern(n):

```
for i in range(1, n+1):  
    for j in range(1, i):  
        print(" ", end = " ")  
    for k in range(i, n+1):  
        print(k, end = " ")  
    print()
```

```
print_pattern(int(input("Enter n:"))))
```

Output

Enter n : 5

1 2 3 4 5
2 3 4 5
3 4 5
4 5
5

f) def print_pattern(n, m):

```
for i in range(1, n+1):  
    for j in range(1, m+1):  
        if (j == 1 or i == n or j == 1 or j == m):  
            print("*", end = " ")  
        else:  
            print(" ", end = " ")
```

print()

r = int(input("Enter row:"))

c = int(input("Enter col:"))

print-pattern(r, c)

Output

Enter row: 5

Enter col: 10

```
* * * * * * * * * *  
* * * * * * * * * *  
* * * * * * * * * *  
* * * * * * * * * *  
* * * * * * * * * *
```

3) def print-pattern(n):

 for i in range(1, n+1):

 for j in range(1, n+1):

 print("*", end="")

 print()

print-pattern(int(input("Enter n:"))))

Output

Enter n: 5

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

```
h> def print_pattern(n):  
    for i in range(0, n+1):  
        for j in range(0, n-i):  
            print(" ", end=" ")  
        for j in range(0, i+1):  
            print("*", end=" ")  
        print()  
print(pattern(int(input("Enter n:"))))
```

Output

Enter n: 4

```
*  
* * *  
* * * * *  
* * * * * * *
```

```
i> def print_pattern(n):  
    for i in range(1, n+1):  
        for j in range(0, i):  
            print(" ", end=" ")  
        for j in range(1, (n*2 - (2*i-1))+1):  
            if i==1 or j==1 or j==(n*2 - (2*i-1)):  
                print("*", end=" ")  
            else:  
                print(" ", end=" ")  
        print()  
print(pattern(int(input("Enter n:"))))
```

~~is def~~ Output

Enter n: 4

A decorative horizontal border at the top of the page, featuring a repeating pattern of small blue asterisks.

```
8> def print_pattern(n):
    for i in range(1, n+1):
        for j in range(0, i):
            print(" ", end=" ")
        for j in range(1, (n*2 - (2*c-1))+1):
            print("* ", end=" ")
        print()
print(pattern(int(input("Enter n:"))))
```

Output

A grid of 15 asterisks arranged in three rows. The first four columns have four asterisks each. The fifth column has three asterisks. The last two columns have two asterisks each. The last row contains only the first five asterisks from the first column.

```

k) def print_pattern(n):
    for i in range(1, n+1):
        for j in range(1, n-i+1):
            print(" ", end="")
        for j in range(1, 2*i):
            if j == 1 or j == 2*i - 1:

```

```
    print("* " , end = " ")
else:
    print(" " , end = " ")
print()
for i in range(n-1, 0, -1):
    for j in range(1, n-i+1):
        print(" " , end = " ")
    for j in range(1, 2*i):
        if j == 1 or j == 2*i-1:
            print("* " , end = " ")
        else:
            print(" " , end = " ")
    print()
```

```
print(pattern(int(input("Enter n"))))
```

Output

Enter n: 4

```
*      *
 *      *
 *      *
*      *
 *      *
 *      *
 *      *
```

```
l> def print-pattern(n):
    for i in range(1, n+1):
        for j in range(1, n-i+1):
            print(" " , end = " ")
        for j in range(1, 2*i):
            if j == 1 or j == 2*i-1:
                print("* " , end = " ")
            else:
                print(" " , end = " ")
    print()
```

```
for i in range (n-1, 0, -1):  
    for j in range (1, n-i+1):  
        print (" ", end = " ")  
    for j in range (1, 2*i):  
        print ("*", end = " ")  
    print ()
```

```
print - pattern (int (input ("Enter n: ")))
```

Output:-

Enter n: 4

```
*  
* * *  
* * * * *  
* * * * * *  
* * * *  
* *  
*
```

m> def print - pattern (n) :

```
for i in range (1, n+1):  
    for j in range (1, i):  
        print (" ", end = " ")  
    for j in range (i, n+1):  
        print ("*", end = " ")  
    print ()
```

```
print - pattern (int (input ("Enter n: ")))
```

Output

Enter n: 5

\$ \$ \$ \$ \$
\$ \$ \$ \$
\$ \$ \$
\$ \$
\$

```
n> def print_pattern(n):  
    for i in range(n, 0, -1):  
        for j in range(1, i):  
            print(" ", end=" ")  
        for j in range(1, n+1):  
            print("# ", end=" ")  
        print()  
print_pattern(5 int(input("Enter n: ")))
```

Output

Enter n:

#

13.

a) import math

def seriesa(x, n):

res = 1

pos = 1

temp = 2

for i in range(1, n):

fact = 1

for j in range(1, temp+1):

fact = fact * j

pos = pos * (-1)

m = pos * math.pow(x, temp)/fact

res = res + m

temp += 2

return res

x = int(input("Enter the value of x:"))

n = int(input("Enter the value of n:"))

print(seriesa(x, n))

Output

Enter the value of x: 10

Enter the value of n: 5

1458 • 9365 ...

b) def series_b(x, n):

res = 1.0

for i in range(n, 0, -1):

res = 1 + x * res / i

return res

x = int(input("Enter the value of x: "))

n = int(input("Enter the value of n: "))

print(series_b(x, n))

Output

Enter the value of x: 1

Enter the value of n: 2

2.5

14>

def palindrome():

num = int(input("Enter a number: "))

temp = num

rev = 0

while (num > 0):

dig = num % 10

rev = rev * 10 + dig

num = num // 10

if (temp == rev):

else return True

return False

print(palindrome())

Output :-

Enter a number: 17871

True

15° def sum_of_digits(n):

 sum=0

 while n>0:

 sum += n%10

 n // = 10

 return sum

n = int(input("Enter n: "))

print(sum_of_digits(n))

Output

Enter n : 154

10

16° def armstrong(n)

 sum=0

 temp=n

 while temp>0

 digit = temp % 10

 sum += digit ** 3

 temp // = 10

 if n == sum :

 return True

else

 return False

```
for i in range(1, 1001):
    if (Armstrong(i)):
        print(i)
```

Output

1
153
370
371
407

Q7. def coprime(m, n):
 gcd = 1
 for i in range(1, m+1)
 if m % i == 0 and n % i == 0
 gcd = i
 return (True if gcd == 1 else False)

m = int(input("Enter number m:"))
n = int(input("Enter number n:"))
print("m and n are co-primes : ", coprime(m, n))

Output

Enter number m : 5
Enter number n: 4
m and n are co-primes : True

18. def multiply(a, b):

 res = 0

 for i in range(1, b+1):

 res += a

 return res

a = int(input("Enter a: "))

b = int(input("Enter b: "))

print("a * b is : ", multiply(a, b))

Output

Enter a: 12

Enter b: 4

a * b is: 48

19.

n = float(input("Enter a value: "))

if n == 0:

 print("Error: The first value cannot be 0.")

 exit()

sum = n

c = 1

while True:

 n = float(input("Enter another value (0 to quit): "))

 if n == 0:

 break

 sum += n

 c += 1

average = sum / c

print ("The average of the values entered is : ", average)

Output

Enter a value : 15

Enter another value (0 to quit) : 22

Enter another value (0 to quit) : 43

Enter another value (0 to quit) : 79

Enter another value (0 to quit) : 0

The average of the values entered is : 39.75

20.

prices = [4.95, 9.95, 14.95, 19.95, 24.95]

discount_rate = 0.6

print (" Price Discount New Price)

for price in prices :

 discount = price * discount_rate

 new = price - discount

 print("{:.2f} {:.2f} {:.2f}" .format
(price , discount , new))

Output

Price	Discount	New Price
4.95	2.97	1.98
9.95	5.97	3.98
14.95	8.97	5.98
19.95	11.97	7.98
24.95	14.97	9.98

```
21. def is-prime(n):  
    if n < 1:  
        return False  
    if n <= 3:  
        return True  
    if n % 2 == 0 or n % 3 == 0:  
        return False  
    for i in range(5, int(n**0.5)+1):  
        if (n % i == 0 or n % (i+2) == 0):  
            return False  
    return True.  
  
def prime-factors(n):  
    divisor = 2  
    while divisor <= n:  
        if (n % divisor == 0):  
            if (is-prime(divisor)):  
                print(divisor)  
            n = n / divisor  
        else:  
            divisor += 1  
  
n = int(input("Enter an integer (2 or greater):"))  
if n < 2:  
    print("Please enter an integer greater than or  
equal to 2.")  
else:  
    print("The prime factors of {} are: {}".format(n))  
    prime-factors(n)
```

Output

Enter an integer (2 or greater) : 72

The prime factors of 72 are:

2
2
2
3
3