**MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE REPUBLIC OF KAZAKHSTAN**

****

**International Information Technology University**

**Department of Cybersecurity**

**PROGRAMMING LANGUAGE PYTHON**

|  |  |
| --- | --- |
| **Student** | **Riash Meirzhan** |
|  | *(given name, surname) (фамилия, имя, отчество)* |
| **Group** | Sis-2214 |
| **Lecturer** | **Nazzere E. Oryngozha** |
| **Аssessment** |  |

Almaty, 2024

**Week 2**

**Module 2. Python Data Types, Variables, Operators, and Basic I/O Operations**

**Task 1**

There is a list a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89].

Print all the elements that are less than 5.

CODE:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

result = [x for x in a if x < 5]

print(result)

**Task 2**

The lists are given:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89];

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

You need to return a list that consists of elements that are common to these two lists.

CODE:

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

common\_elements = list(set(a) & set(b))

print(common\_elements)

**Task 3**

Sort the dictionary by value in ascending and descending order.

Import the necessary module and declare the dictionary:

import operator

d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

CODE:

import operator

d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

ascending\_order = dict(sorted(d.items(), key=operator.itemgetter(1)))

descending\_order = dict(sorted(d.items(), key=operator.itemgetter(1), reverse=True))

print("Ascending:", ascending\_order)

print("Descending:", descending\_order)

**Task 4**

Write a program to merge several dictionaries into one.

Let's say here are our dictionaries:

dict\_a = {1:10, 2:20}

dict\_b = {3:30, 4:40}

dict\_c = {5:50, 6:60}

CODE:

dict\_a = {1: 10, 2: 20}  
dict\_b = {3: 30, 4: 40}  
dict\_c = {5: 50, 6: 60}  
  
merged\_dict = {}  
  
for d in (dict\_a, dict\_b, dict\_c):  
 merged\_dict.update(d)  
  
print("Merged dictionary:", merged\_dict)

**Task 5**

Find the three keys with the highest values in the dictionary my\_dict = {'a':500, 'b':5874, 'c': 560,'d':400, 'e':5874, 'f': 20}.

CODE:

my\_dict = {'a':500, 'b':5874, 'c':560, 'd':400, 'e':5874, 'f':20}

top\_three = sorted(my\_dict, key=my\_dict.get, reverse=True)[:3]

print(top\_three)

**Task 6**

Write a code that converts an integer into a string, despite the fact that it can be used in any number system.

CODE:

number = 1234

number\_str = str(number)

print(number\_str)

**Task 7**

You need to print the first n lines of Pascal's triangle. In this triangle, there are units at the top and on the sides, and each number inside is equal to the sum of the two numbers above it.

CODE:

def pascals\_triangle(n):

triangle = [[1]]

for i in range(1, n):

row = [1]

for j in range(1, i):

row.append(triangle[i-1][j-1] + triangle[i-1][j])

row.append(1)

triangle.append(row)

return triangle

n = 5

triangle = pascals\_triangle(n)

for row in triangle:

print(row)