Test A III - UAI 735I

## Exam – written part – winter 2020/2021

|  |  |
| --- | --- |
| Name of student |  |
| Date |  |

# 1. What is result 3 points

|  |
| --- |
| >>> list\_of\_elements ={'0': 0, '1': 1,'2': 2,'3': 3,'4': 4,'5': 5,'6': 6,'7': 7,'8': 8,'9': 9, 'A': 10, 'B': 11, 'C': 12, 'D': 13, 'E': 14, 'F': 15, 'G': 16, 'H': 17, 'I': 18,'J': 19, 'K': 20,'L': 21,'M': 22,'N': 23,'O': 24,'P': 25,'Q': 26,'R': 27,'S': 28,'T': 29,'U': 30,'V': 31,'W': 32,'X': 33,'Y': 34,'Z': 35} |
| >>> l={k.lower():v  for k,v in list\_of\_elements.items() if v > 5 and k.isdigit()}  >>> print(l) # print all |

# 2. Format 4 points

|  |
| --- |
| >>> m = ({1:{1,2,3},2:[9,8,0],8:[11,15.55]},())  >>> print('{1},{0} hello {3:.5f}{2}'.format(m[1],m[0][2],m[0][2],m[0][8][1])) |
| >>> Output is: |
| >>> m = ({1:{1,2,3},2:[9,8,0],8:{11,15.55}},())  >>> print('{1},{0} hello {3:.5f}{2}'.format(m[1],m[0][2],m[0][2],m[0][8][1])) |
| >>> Output is: |

3.What the function lambda is doing 4 points

|  |
| --- |
| import random as rd  x={rd.randint(10,1000) for i in range(1000)}  print(list(filter(lambda u: all([(u % i or u==i) for i in range(2,u+1)]), x)))) |
| # write the same with *def* |
| Describe what this lambda function doing |

# 4 Draw a simple activity diagram 3 points

|  |
| --- |
| Draw a simple aktivity diagram which will   * Read words from a text file * Decide if the word contains only ascii letters * If yes * Find out the spanish translation in the database * If it is a digit, print it to a screen * Print the pair into output text file |
|  |
|  |

# 5 What are outputs and types 2 points

|  |
| --- |
| def call(x,y=10):  if y < x:  return  return y - x |
| print(call(9,8)) #what is output  print(call(9)) #what is output |

# 6 What is output 1 point

|  |
| --- |
| >>> def abc(x):  try:  print(20/x)  except:  print("Wrong ")  finally:  print("Cheers ") |
| >>>abc(0)  >>>abc(10) |

# 7 Answer 1 point

|  |
| --- |
| What is not a core data structure in Python?  -list  -module  -dictionary  -tuple |
|  |

# 8 What is result 1 point

|  |
| --- |
| >>> y=lambda x: x\*4  >>> import math |
| >>> print(int(y(pi))) |

9.Describe 4 points

|  |
| --- |
| def popular\_words(s,l):  d={}  l0=s.replace('\n',' ').lower().split()  for i in l:  if i in l0:  d[i] = l0.count(i)  else:  d[i] = 0  return d  popular\_words('''  When I was One  I had just begun  When I was Two  I was nearly new  ''', ['i', 'was', 'three', 'near']) |
| # output |
| # describe  What the program is doing    Is the program right or is there a mistake? If yes, what error we will see…..    How the method will be called to print readable output on a screen |

# 10. Work with files 2 points

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # explain the symbols for opening the files   |  |  | | --- | --- | | r |  | | rb |  | | r+ |  | | rb+ |  | | w |  | | wb |  | | w+ |  | | wb+ |  | | a |  | | ab |  | | a+ |  | | ab+ |  | |

# 11 What is output 3 points

|  |
| --- |
| def call(v1=20,v2=5,v3=2):  if v3 > 2:  print(v1\*v2-v3)  else:  print(v3)  call(7,4,1) |
| # what is output |

# 12 What is the difference 2 points

|  |
| --- |
| 1>>> import math as m  2>>> from math import pi |
| # write d = 2 \* pi \* 5 in both cases  1>>>  2>>> |

# 13. What is output 3 points

|  |
| --- |
| >>> x="Java Python, Ruby"  >>> x[:11].endswith("n") |
|  |
| >>> x.index("y") |
|  |
| >>> x[::-1].index("y") |
|  |

# 14. what is wrong, (if any) and why 3 points

|  |
| --- |
| a=[1,5,4,7,8,5,7,8]  dic={k:v for k,v in enumerate(a)}  print(dic[5]) # 1  print(dic[8]) # 2  print(dic.items()) # 3 |
| Wrong or right ? |
| from functools import reduce  x = [1, 2, 3, 4]  print(-8 == reduce((lambda x, y: x - y), [1, 2, 3, 4])) |
| Wrong or right ? |
| print(os.listdir()) |
| Wrong or right ? |

# 15. What the program is doing? 2 points

|  |
| --- |
| d3= [("x",3, "y",3),("x",2, "y",-1),("x",2, "y",3),("x",-1, "y",0),("x",2, "y",2),("x",3, "y",-1),("x",1, "y",2),("x",0, "y",2),("x",1, "y",0),("x",0, "y",-1),("x",0, "y",1),("x",-1, "y",0),("x",1, "y",-1),("x",-1, "y",1),("x",2, "y",0),("x",1, "y",2),("x",-1, "y",0),("x",1, "y",2),("x",-1, "y",3),("x",-1, "y",2),("x",0, "y",0),("x",1, "y",3),("x",2, "y",0),("x",0, "y",2),("x",-1, "y",-1),("x",0, "y",2),("x",2, "y",3),("x",2, "y",-1),("x",2, "y",0),("x",3, "y",3),("x",-1, "y",0),("x",2, "y",-1),("x",0, "y",-1),("x",2, "y",2),("x",-1, "y",0),("x",2, "y",-1),("x",0, "y",1),("x",0, "y",3),("x",3, "y",1),("x",0, "y",2),("x",2, "y",0),("x",0, "y",2),("x",1, "y",2),("x",0, "y",3),("x",1, "y",0),("x",-1, "y",0),("x",-1, "y",3),("x",1, "y",-1),("x",0, "y",2),("x",2, "y",3),("x",-1, "y",1),("x",1, "y",0),("x",2, "y",3),("x",2, "y",2),("x",1, "y",2),("x",0, "y",-1),("x",-1, "y",1),("x",1, "y",0),("x",0, "y",2),("x",3, "y",3),("x",2, "y",-1),("x",1, "y",3),("x",2, "y",3),("x",1, "y",2),("x",3, "y",0),("x",0, "y",2),("x",3, "y",0),("x",0, "y",-1),("x",3, "y",3),("x",1, "y",3),("x",-1, "y",-1),("x",1, "y",1),("x",0, "y",3),("x",3, "y",-1),("x",1, "y",2),("x",1, "y",0),("x",-1, "y",0),("x",2, "y",3),("x",1, "y",-1),("x",0, "y",0),("x",0, "y",1),("x",1, "y",1),("x",-1, "y",1),("x",-1, "y",0),("x",2, "y",-1),("x",1, "y",-1),("x",3, "y",-1),("x",2, "y",1),("x",0, "y",1),("x",1, "y",1),("x",3, "y",1),("x",2, "y",3),("x",1, "y",3),("x",1, "y",3),("x",1, "y",-1),("x",2, "y",1),("x",0, "y",1),("x",3, "y",0),("x",1, "y",-1),("x",3, "y",-1)]  newSet = set()  for i in d3:  newSet.add(i)  print(newSet) |
| # what the program is doing - describe |