Exam Practice

Refer to Python documentation in the appendix.

1 a) Given the following function:

```
1. def subtract(x, y):
2.   z = 0
3.   while x > y:
4.     x -= 1
5.     z += 1
6.   print(z)
```

i) The function `subtract` subtracts 2 numbers passed as parameters. Explain why the above function does not work for all integer values of x and y.

ii) Modify the function such that it prints the correct result for all integer values of x and y passed as parameters. The statements from line 3 to 5 must remain intact and there should be no additional loops.

iii) A prime number is a number that is divisible by 1 and itself. 1, 2, 3, 7, 11 are examples of prime numbers. A simple algorithm to check if a number N is prime is by checking whether N is divisible by any consecutive numbers starting from 2 to N-1. If any of these numbers is divisible, then N is not a prime. Write a function `isPrime(n)` that has an integer parameter and returns `True` if n in prime and `False` otherwise.

1 b) Given the following function:

```
1. def f(s):
2.   n = 0
3.   for c in range(len(s)):
4.     if int(s[c]) % 2 == 0:
5.       n+=1
6.   return n
```

i) Explain what the function `f` does.

ii) What is the output if the function is called as follows:

```
f('162534')
```

iii) Rewrite the `for…range` loop using a while loop.

1 c) Given the following function:

```
1. def printList(param):
2.    for n, v in enumerate(param):
3.        print(n+1, v)
```

i) When the `printList` function is called with a parameter value, the output shows:

1 one
2 two
3 three

Show the contents of the parameter param that produces the above printout.

ii) Rewrite the for loop without the `enumerate()` function. It should produce the same result.

2 a) Write a function `isBinary(digits)` that checks whether the digits passed as a string parameter is 0 or 1. The function returns `True` if all the digits are either 0 or 1, and `False` otherwise.

2 b) An example of converting a binary number 1011 to decimal is as follows:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11$$

Write a function `toDecimal(binary)` that has binary digits as a string parameter and returns the decimal value.

2 c) Write a main function that continually prompts for a binary string and displays the decimal number equivalent. Display 'Invalid binary number' if the input string does not consists of 0 and 1. The program ends when 0s are entered. Before the program ends, display the number of binary numbers converted. The following is a sample program execution:

```
Enter binary number: 100
100 binary is decimal 4
Enter binary number: 123
Invalid binary number
Enter binary number: 1110
1110 binary is decimal 14
Enter binary number: 0
2 binary numbers converted.
```

The main function must make use of the functions written is part a) and b).

3 a) Suggest an appropriate data structure, if necessary, for the following program requirements by giving examples of how the data structure would look like with sample values. Briefly justify the use of the data structure.

i) A dice is rolled multiple times. For each roll, the dice value is printed. Program stops when the dice value is 6. A count of how many times it took to reach value 6 is printed.

ii) The hours worked for a daily rated employee is entered for 5 days. After input for 5 days, the program displays a summary the hours worked and pay for each of the 5 days.

iii) An airport has 4 terminals. The program reads and stores the arrival and departure number of visitors for each terminal. A summary of the arrival and departure numbers by terminal is printed.

iv) A dice game is played with multiple players, over many rounds. Player names are recorded. For each round, players take turn to play, until there is a winner. The winning count is recorded for that player. A overall summary of the number of rounds won by each player is displayed when the game ends.

b) A single player dice game is played many rounds. For each round, the player throws a dice multiple times until the dice value is 6. The game ends when the player throws a value 6 on the first try for that round. This is how the data structure looks like after a game that lasted 4 rounds:

score=[[1,6], [2,3,1,6], [3,5,6], [6]]

Base on the above data structure, write code to print the summary score as follows:

R1 1 6
R2 2 3 1 6
R3 3 5 6
R4 6
Average number of rolls per round = 2.5
(obtained by taking the total number of rolls divided by the number of rounds)

4. Write a program that helps a football association track the scores of matches in a football season. The current number of wins, draws, loses for all teams are stored in a file `scores.txt`. Assume there are only 8 teams, A to H. A sample of the file in the mid-season is as follows:

A 24 1 0
B 16 3 6
C 15 4 6
D 12 5 8
E 10 7 8
F 9 9 7
G 9 8 8
H 8 11 6

a) Write a function `readScoresFromFile(filename)` that has the scores stored in a file as parameter. The function populates a dictionary structure and returns the dictionary. The populated dictionary from the file should be as follows:

```
scoredict={'A':[24,1,0],'B':[16,3,6],'C':[15,4,6],…,'H':[8,11,6] }
```

b) Write a function `inputScore(scoredict)` that has the score dictionary as parameter. The function reads in the scores for 1 game and updates the win, draw, lose count of each team. A sample input is as follows:

```
Enter score: A 2 B 1
```

Input is separated by blanks. Update the win count of team A by 1 and the lose count of team B by 1. It is possible that the team name may be entered wrongly. Display an error message and do not update any scores. The following example shows such a scenario:

```
Enter score: A 2 J 1
Team J does not exists. No scores recorded
```

c) Write a function `printScoreTable(scoredict)` that has the score dictionary as parameter. The function prints the score table as follows:

Team	Pts	W	D	L
A	73	24	1	0
B	51	16	3	6
C	49	15	4	6

…

A win is 3 points, draw 1 point and lose 0 points. There is no requirement to sort the output by team with the highest points.

d) Write a function `saveScoresToFile(scoredict)` that has the score dictionary as parameter. The function writes the scores into scores.txt in the format as shown earlier.

e) Write a main function to run the program. The readScoresFromFile() function is called to read in the current score states from scores.txt. This is followed by the following menu which will call the respective functions written in part b and c:

Menu
1. Input score
2. Print score table
3. Exit

Enter option:

When option 3 to exit is entered, call save`ScoresToFile()` function to write the scores to `scores.txt` file.

Appendix

Method	Meaning
<dict>.keys()	Returns a sequence of keys.
<dict>.values()	Returns a sequence of values.
<dict>.items()	Returns a sequence of tuples (key, value) representing the key-value pairs.
<key> in <dict>	Returns true if dictionary contains the specified key, false if it doesn’t.
for <var> in <dict>:	Loop over the keys.
<dict>.get(<key>, <default>)	If dictionary has key, returns its value; otherwise returns default.
del <dict>[<key>]	Deletes the specified entry.
<dict>.clear()	Deletes all entries.
<file>.read()	Returns the unread content as a single string
<file>.readline()	Returns the next line of the file.
<file>.readlines()	Returns a sequence (a list) of unread lines in the file.
<file>.write(str)	Writes string to the file, and return the number of characters.
<file>.close()	Closes file and release resources
open(filename, filemode)	Opens file
<list>.append(item)	Adds item at end of list
<list>.insert(pos, item)	Adds item at specified position of list
<list>[pos] = value	Replaces element at pos with value
<list>[start:end] = sequence	Replaces elements at pos start to end -1 with elements in sequence
<list>.remove(item)	Removes item in list
<list>.pop(pos)	Removes item at pos in list
<list>.clear()	Removes all items in list
list(sequence)	Converts sequence to list
<list>.sort()	Sorts (order) the list. A comparison function

	may be passed as a parameter.
<list>.reverse()	Reverses the list.
<list>.index(x)	Returns index of first occurrence of x.
<list>.count(x)	Returns the number of occurrences of x in list.