

CSC1015F Assignment 4B (70 Marks)

Control (if, for, while)

Assignment Instructions

This assignment involves constructing Python programs that use input and output statements, 'if' and 'if-else' control flow statements, 'for' statements, 'while' statements and statements that perform numerical manipulation.

There are three questions in this assignment. All the three questions are effectively dependent on each other. The ultimate goal is to write the program required in question 3. Applying strategies of simplification and divide-and-conquer, the first two questions are sub problems of question 3.

NOTE Your solutions to this assignment will be evaluated for correctness. Furthermore, your solutions will also be evaluated for the following qualities:

- Documentation
 - Use of comments at the top of your code to identify program purpose, author and date.
 - Use of comments within your code to explain each non-obvious functional unit of code.
- General style/readability
 - The use of meaningful names for variables and functions.
- Algorithmic qualities
 - Efficiency, simplicity

These criteria will be manually assessed by a tutor and commented upon. In this assignment, up to 10 marks will be deducted for deficiencies.

Question 1 [20 marks]

Write a program called 'row.py' that asks the user to enter a number, n , where $-6 < n < 93$. The program will print a sequence of 7 numbers, starting from that value.

Output will take the following form:

```
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
```

Numbers are printed with exactly two characters. ('[d]' represents an optional digit.)

When a number is greater than 9, both characters are digits. When a number is less than ten and greater than zero the first character is a space. When the number is negative, the first character is a minus sign, '-'.

To put it more formally, numbers are printed with a field width of two and are right-justified.

Adjacent numbers are separated by a single space. There are no spaces after the final number.

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 93:

12

12 13 14 15 16 17 18

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 93:

-7

Invalid input! The value of 'n' should be between -6 and 93.

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 93:

95

Invalid input! The value of 'n' should be between -6 and 93.

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 93:

0

0 1 2 3 4 5 6

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 93:

7

7 8 9 10 11 12 13

HINTS:

- Use an 'if' statement to see if a number needs padding with a leading space.
- Use the print statement 'sep' and 'end' parameters to help control spacing and line ends.
- To get the spacing between numbers right, you should treat the first number as a separate before looping for subsequent numbers i.e. get the first number and print, then for each subsequent number, print a space then the number.

Question 2 [20 marks]

Write a program called 'column.py' that asks the user to enter a number, n , where $-6 < n < 2$.

Starting from n , the program will print out every 7th number in the range n to $n+41$.

Output will take the following form:

```
[d]d
[d]d
[d]d
[d]d
[d]d
[d]d
```

That is, numbers are printed using a field width of 2 and are right-justified. ('[d]' represents an optional digit.)

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 2:

-5

-5

2

9

16

23

30

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 2:

-7

Invalid input! The value of 'n' should be between -6 and 2.

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 2:

4

Invalid input! The value of 'n' should be between -6 and 2.

Question 3 [30 marks]

Write a program called 'grid.py' that accepts a number, n , where $-6 < n < 2$. The program will print out the numbers n to $n+41$ as 6 rows of 7 numbers. The first row will contain the values n to $n+6$, the second, the values $n+7$ to $n+7+6$, and so on.

Output will take the following form:

```
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
[d]d [d]d [d]d [d]d [d]d [d]d [d]d
```

That is, numbers are printed using a field width of 2, and are right justified. ('[d]' represents an optional digit.) Fields are separated by a single space. There are no spaces after the final field.

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 2:

-5

-5 -4 -3 -2 -1 0 1

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 31 32 33 34 35 36

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a number between -6 and 2:

6

Invalid input! The value of 'n' should be between -6 and 2.

HINT: Use a 'for' loop within a 'for' loop.

Submission

Create and submit a Zip file called 'ABCXYZ123.zip' (where ABCXYZ123 is YOUR student number) containing `row.py`, `column.py`, and `grid.py`.

NOTES:

1. FOLDERS ARE NOT ALLOWED IN THE ZIP FILE.
2. As you will submit your assignment to the Automarker, the Assignment tab may say something like "Not Complete". THIS IS COMPLETELY NORMAL. IGNORE IT.