CSC1015F Assignment 3: 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, '`while`' statements, '`for`' statements, and statements that perform numerical manipulation.

Question 1 [10 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
```

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:

```
Enter the start number: 12
12 13 14 15 16 17 18
```

Question 2 [10 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 I/O:

```
Enter a number: -5
-5
 2
 9
16
23
30
```

CONTINUED

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 I/O:

```
Enter the start number: -2
-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 37 38 39
```

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

Question 4 [20 marks]

Write a program called 'printmonth.py' that asks the user for a month name and start day and then prints the calendar for that month in a 6 row by 7 column grid. (Ignoring issues of leap years, assume February has 28 days).

Sample I/O:

```
Enter the month('January', ...,'December'): November
Enter the start day ('Monday', ..., 'Sunday'): Sunday
November
Mo Tu We Th Fr Sa Su
                   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
```

Note (for the sake of automatic marking) that output must always form a 6 row by 7 column grid. Spaces should be used where necessary.

HINT: Draw on your answer to the previous question. Calculate the start value based on the day, print spaces for numbers less than one and for numbers greater than the number of days in the month.

CONTINUED

Question 5 [30 marks]

Write a program called 'palindromeprime.py' that finding all palindromic primes between two integers supplied as input (start and end points are excluded).

A palindrome number is a number that reads the same from the front and the back. Examples are: 212, 44, 9009, 4567654. To calculate whether a number is a palindrome or not, you can first reverse the number (using the % operator and a loop, or a String) and then check for equality.

A prime number is one that is only divisible by 1 and itself. Examples are: 3, 11, 313.

Some examples of palindromic primes are: 11, 191, 313

Sample I/O:

```
Enter the start point N:
200
Enter the end point M:
800
The palindromic primes are:
313
353
373
383
727
757
787
797
```

Submission

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

END