

Introduction to Programming Homework 1 Solutions

Note : There are (many) different ways to answer many of these questions.

Exercise 1

- a. Create a variable called `desc` with the string value 'The sum of 2 plus 2 is equal to'.

In []:

```
desc = 'The sum of 2 plus 2 is equal to'
```

- b. Check the type of `desc`.

In []:

```
type(desc)
```

- c. Call the command `print(desc, 2+2)`. Notice that you are giving the `print` function two input variables. In the context of programming, input variables are usually called *arguments*.

In []:

```
print(desc, 2+2)
```

- d. Create a variable called `print_arg_exp` where you explain in words what `print` would do if you gave it three arguments instead of two. For example, a *wrong* answer would be :
`print_arg_exp = 'print would crash if given three arguments'.`

In []:

```
print_arg_exp = 'print will display the value of each argument separated by a space'
```

Exercise 2

Write a one line command to output the list of values modulo 373 of the numbers 1 to 999 inclusive. Hint : use list comprehension.

In []:

```
print([ x % 373 for x in range(1,1000) ])
```

Exercise 3

Start by creating the lists :

```
days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
months = ['January', 'February', 'March', 'April', 'May',
'June', 'July', 'August', 'September', 'October', 'November', 'December'
]
```

Write a one line command to output the list of pairs (lists in this case)

```
[[ 'January', 31], [ 'February', 28], [ 'March', 31],
[ 'April', 30], [ 'May', 31], [ 'June', 30], [ 'July', 31],
[ 'August', 31], [ 'September', 30], [ 'October', 31],
[ 'November', 30], [ 'December', 31]]
```

We will learn how to properly *fold* (or intertwine) lists together later.

In []:

```
days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
months = ['January', 'February', 'March', 'April', 'May',
'June', 'July', 'August', 'September', 'October', 'November', 'December']

print([ [months[i], days[i]] for i in range(len(days)) ])
```

Exercise 4

Use the `sum()` to approximate π using the Bailey–Borwein–Plouffe formula

$$\pi = \sum_{k=0}^{\infty} \left[\frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right) \right]$$

Use as many summands as you feel appropriate.

In []:

```
sum([ (4/(8*k+1)-2/(8*k+4)-1/(8*k+5)-1/(8*k+6))/16**k for k in range(0,100) ])
```

Exercise 5

When using list comprehension, you can also filter a list using the `if` keyword. The `if` keyword must be followed by an expression that returns a Boolean value. For example

```
[ x for x in range(1, 100) if x > 50 and x % 3 == 1 ]
```

For `x in range(1, 100)`, `x` is added to the list if `x > 50` and `x % 3 == 1` is `True`. This should remind you of normal set notation

$$\{x \mid x \in \{1, 2, \dots, 99\}, x > 5, \text{ and } x = 1 \bmod 3\}$$

- a. Find all even numbers between 1000 and 3000 inclusive which are divisible by 7 but are not a multiples of 3.

In []:

```
print([ x for x in range(1000,3001,2) if x % 7 == 0 and not x % 3 == 0 ])
```

- b. Use python to verify that

$$\{x \mid x \in \{1, 2, \dots, 10000\}, \text{ and } x^2 = 2 \bmod 5\} = \emptyset$$

In []:

```
print([ x for x in range(1,10001) if x**2 % 5 == 2 ])
```