# Introduction to Programming and Data Structures

## Python – Basic Input/output

Malay Bhattacharyya

Associate Professor

MIU, CAIML, TIH
Indian Statistical Institute, Kolkata

August, 2023

# Standard Input/Output functions

**I/O from the terminal:**

Value can be printed without mentioning the type as:

```
print(*obj,sep=' ',end='\n',file=sys.stdout,flush=False)
```

Value is taken in a string and can be converted to appropriate type using int(), float(), bool(), etc. as:

```
input([prompt])
```

**I/O from files:**

- `open()`, `close()` # Files are opened in r/w/a mode and the address is returned to a file pointer
- `read()`, `write()` # With a file pointer
- `readline()` # With a file pointer
- `readlines()`, `writelines()` # With a file pointer

# print() – Special features

```
a, b, c = 1, 2, 3
print(a, b, c, sep = '')
print(a, b, c, sep = '-')
print(a, b, c, sep = '1.1')
x = print('Hi')
print(x)
print(type(x))
```

# print() – Special features

```
a, b, c = 1, 2, 3
print(a, b, c, sep = '')
print(a, b, c, sep = '-')
print(a, b, c, sep = '1.1')
x = print('Hi')
print(x)
print(type(x))
```

**Output:**

```
123
1-2-3
11.121.13
Hi
None
<class 'NoneType'>
```

# print() – Special features

```
a = 7
print(a, 'is prime', end = ';') # ',' includes a space
b = 'prime'
print('7 is ' + b, end = ';') # '+' works only on strings
```

# print() – Special features

```
a = 7
print(a, 'is prime', end = ';') # ',' includes a space
b = 'prime'
print('7 is ' + b, end = ';') # '+' works only on strings
```

**Output:**

```
7 is prime;7 is prime;
```

# print() – Special features

```
ls = [[1, 2, 3], [4, 5, 6]]
for i in range(len(ls)):
    for j in range(len(ls[0])):
        print(ls[i][j], end = ' ')
    print()
```

# print() – Special features

```
ls = [[1, 2, 3], [4, 5, 6]]
for i in range(len(ls)):
    for j in range(len(ls[0])):
        print(ls[i][j], end = ' ')
    print()
```

**Output:**

```
1 2 3
4 5 6
```

# print() – Special features

```
inputFile = open('test.txt', 'w')
print('Write your own fate!!!', file = inputFile)
inputFile.close()
```

# print() – Special features

```
inputFile = open('test.txt', 'w')
print('Write your own fate!!!', file = inputFile)
inputFile.close()
```

**Output:**

```
Write your own fate!!!
```

- written within test.txt.

# input() – Special features

```
x = int(input())
print(x)
n = input('Enter three integers: ')
print(n, list(n))
n1, n2, n3 = input('Enter three integers: ').split()
print(n1+n2+n3, int(n1)+int(n2)+int(n3))
```

# input() – Special features

```
x = int(input())
print(x)
n = input('Enter three integers: ')
print(n, list(n))
n1, n2, n3 = input('Enter three integers: ').split()
print(n1+n2+n3, int(n1)+int(n2)+int(n3))
```

**Output:**

```
10
10
Enter three integers: 1 2 3
1 2 3 ['1', ' ', '2', ' ', '3']
Enter three integers: 1 2 3
123 6
```

# input() – Special features

```
r = int(input('Enter the number of rows: '))
c = int(input('Enter the number of columns: '))
MAT = [[int(input()) for i in range(c)] for j in range(r)]
print(MAT)
```

# input() – Special features

```
r = int(input('Enter the number of rows: '))
c = int(input('Enter the number of columns: '))
MAT = [[int(input()) for i in range(c)] for j in range(r)]
print(MAT)
```

**Output:**

```
Enter the number of rows: 2
Enter the number of columns: 3
1
2
3
4
5
6
[[1, 2, 3], [4, 5, 6]]
```

# `input()` – Special features

Explore the `reshape()` function!!!

# Reading data from file

```
def read(file):
    f = open(file, 'r')
    output = f.read()
    f.close()
    return output
output = read('Data.txt')
```

# Reading data from file

```
def read(file):
    f = open(file, 'r')
    output = f.read()
    f.close()
    return output
output = read('Data.txt')
```

**Reading data from file (alternative approach):**

```
with open('Data.txt', 'r') as f: output = f.read();
```

# Special Input/Output functions

**Reading data from a CSV file:**

```
import pandas as pd # Import pandas
pd.read_csv("file.csv") # reading CSV file
```

# Special Input/Output functions

**Reading data from a CSV file:**

```
import pandas as pd # Import pandas
pd.read_csv("file.csv") # reading CSV file
```

**Reading data from an XLS file:**

```
import pandas as pd # Import pandas
pd.read_excel("file.xls") # supports old XLS file formats
pd.read_excel("file.xls", engine='openpyxl') # new formats
```

# Special Input/Output functions

**Reading data from a CSV file:**

```
import pandas as pd # Import pandas
pd.read_csv("file.csv") # reading CSV file
```

**Reading data from an XLS file:**

```
import pandas as pd # Import pandas
pd.read_excel("file.xls") # supports old XLS file formats
pd.read_excel("file.xls", engine='openpyxl') # new formats
```

**Note:** The Python library `openpyxl` must be used to read/write Excel 2010 xlsx files.

# Problems

1. Suppose there are two separate files containing sufficiently large integer values. Write a program that will take those two filenames as user inputs and return the addition result.
   **Note:** An efficient implementation will not depend on the primary memory of the system.
2. Write a program to multiply a pair of matrices given as user input. Keep a check whether the input matrices are multipliable or not.