

1. Suppose we wanted to randomly roll a 6-sided die to get a value between 1 and 6. Assuming we have already imported the random module as:

```
import random
```

How would you use each of the following to generate a die value between 1 and 6?

a) `random.randrange(`

b) `random.randint(`

c) `random.random(`

d) `random.uniform(`

2. Write a function to calculate the distance between two points (x_1, y_1) and (x_2, y_2) by the formula:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

3. Complete the following table about RAM (main memory) and harddisk (secondary storage).

	RAM (main memory)	Harddisk (secondary storage)
Usage - what is stored in it		
Size - how much storage is there on a typical desktop		
Speed - how fast is the information accessed		
Organization - how is information organized		

Below is a summary of the important file operations in Python.

File Operations in Python		
General syntax	Example	Description
open(filename) open(filename, mode)	f = open('data.txt', 'w')	Modes: 'r' read only; 'w' write only; 'a' append; 'r+' both reading and writing. On Windows and Macs, 'b' appended to the mode opens the file in binary mode. Default mode is 'r'
f.close()	f.close()	Close the file to free up system resources.
f.read()	all = f.read()	Returns the whole file as a string.
f.read(size)	chunk = f.read(100)	Returns a string of at most 100 (size) bytes. If the file has been completely read, an empty string is returned.
f.readline()	nextLine = f.readline()	Returns the next line from the file. The newline ('\n') character is left at the end of the string, unless it is the last line of a file which does not end in a newline character.
f.readlines()	allLines = f.readlines()	Returns a list containing all the lines of the file.
f.readlines(size)	someLines = f.readlines(5000)	Returns the next 5000 bytes of line. Only complete lines will be returned.
f.write(string)	f.write('cats and dogs')	Writes the string to the file.
loop over the file object	for line in f: print line,	Memory efficient, fast and simple code to loop over each line in the file.
f.tell()	f.tell()	Returns the position from the beginning of the file in bytes.
f.seek(offset)	f.seek(5000)	Move the file pointer to 5000 bytes from the beginning of the file.
f.seek(offset, from_what)	f.seek(500, 1)	from_what modes: 0 is from the beginning of the file, 1 is from the current position, 2 is from the end of the file.
pickle.dump(x, f)	pickle.dump(x, f)	Converts Python object x to a string representation and writes it to the file.
pickle.load(f)	x = pickle.load(f)	"Unpickles": returns the Python object from file f.

4. Write a program to read a text file "myData.txt" and echo it to the screen.