

Introduction to Python Programming Exercises

Ali Kashefi
`kashefi@stanford.edu`

Problem A1. Does the concept of a “prototype” for functions exist in Python? Based on our class discussion, if it does, what is the primary benefit of this concept? If not, what is the most negative consequence of its absence? You need to mention only one benefit or only one negative consequence. If you write more than one, you receive zero points.

Please write your answer in the following box.

Problem A2. In Python, `int` is an immutable type; however, we can write, for instance, `x = 5`, and then `x += 1`, as shown in the code below. It appears that we can change `x`, even though it is immutable. Based on our class discussion, how can this paradox be resolved?

```
x = 5
x += 1
print(x)
```

Please write your answer in the following box.

Problem A3. In the following box, write the function `f(x)` in Python, which takes a string of digits ranging from 0 to 9 as `x`. It should return `True` if the string is a palindrome and `False` otherwise. A string is a palindrome if it reads the same forwards and backward, such as 12321 or 4554. For this problem, the use of loops is NOT allowed. If any loop is used in your solution, you will receive zero points.

```
def f(x):  
    #complete this function
```

Problem B1. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
print([1] in [1, 2, 3])
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B2. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
a = [1,2,3]
b = a
c = b
b.append(4)
print(a)
print(c)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B3. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
x = 5.0
print(x.is_integer())
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B4. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
a = {"a":1,"b":2}
b = {"a":1,"b":2}
print(a is not b)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B5. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [7, [4, 5, 1], 2]
L.sort()
print(L)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B6. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [2, 3, 5, 7, 11]
print(L[::-8])
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B7. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [3,2]
T = {"a":3, "b":2}
T[L] = 1
print(T)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B8. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
name = "abcdefgh"
print(name[name.find("i")])
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B9. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
for item in range(6,2,-2):  
    print(item,end = " ")
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B10. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
g = lambda x,y=1: x+y  
  
print(g(2,g(2)))
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B11. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
a = "abcdE"  
b = "abcdE"  
print(a is not b)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B12. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [[3,2,7],[4,5,1]]  
L.sort()  
print(L)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B13. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
name = "abcdefgh"
print(name[3:6:-1])
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B14. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [[2,3,1],[4,5,1],7]
L[1].sort()
print(L)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem B15. Write the output of the Python code.

```
G = (n ** 2 for n in range(3))  
print(*G)  
print(*G)
```

Your answer:

Problem B16. Write the output of the Python code.

```
def g(x,y):  
    return x+y  
  
for val in map(g, range(3), [-2,-1,0]):  
    print(val)
```

Your answer:

Problem C1. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
g = lambda x: 1 if x == 1 else g(x-1)

print(g(5))
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C2. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(*K):  
    sum = 0  
    for item in K:  
        sum += item  
    return sum  
  
print(f(1,2,3,f(2,3),f(3)))
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C3. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [{"e",12,-30}, {"b",8,5}, {"c",1,3}]
B = sorted(L, key = lambda x:x[1]+x[2])
print(B)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C4. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(x):  
    x += "E"  
    return  
  
def g(x):  
    x.append("E")  
    return  
  
x1 = "abcd"  
f(x1)  
  
x2 = [x1]  
g(x2)  
  
print(x2)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C5. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def compute(x = 2, y = 6):  
    return y - x  
  
print(compute(y = 4))
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C6. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(**K):  
    return K["a"]  
  
print(f(a=f(a=2,b="b"),b="b",c="c",d=2))
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C7. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
L = [val if val % 3 else -val for val in range(7) if val % 2]
print(L)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C8. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def main():  
    print("A")  
  
if __name__ == "__main__":  
    main()  
  
def main():  
    print("B")  
  
if __name__ == "__main__":  
    main()  
  
if __name__ == "__main__":  
    main()
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C9. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(**K):  
    return K["a"]  
  
print(f(a=1,b=2,b=2,c=3))
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C10. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(x, y):  
    sum = x + y  
    subtraction = x - y  
    multiplication = x*y  
    flag = None  
    if subtraction < 0:  
        flag = True  
    return sum, subtraction, multiplication, flag  
  
num1, num2, _, _ = f(3, 4)  
print(_)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem C11. Write the output of the Python code.

```
class Movie():
    location = "CA"

    def __init__(self, genre):
        self.genre = genre

def main():

    A = Movie("Fiction")
    B = Movie("Romance")

    Movie.location = "NY"
    B.location = "TX"

    print(A.location)
    print(B.location)
    print(A.__dict__)
    print(B.__dict__)

if __name__ == "__main__":
    main()
```

Your answer:

Problem D1. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def g(x,L=[]):  
    L.append(x)  
    return L  
  
def f(x, L=None):  
    if L is None:  
        L = []  
    L = g(x)  
    return L  
  
L = f(2)  
L = f(3)  
print(L)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem D2. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(x):  
    if x == 2:  
        print(x)  
    else:  
        print(x-1)  
  
f(4)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem D3. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def f(x,y,z):  
    return x+y+z  
  
def f(x,y):  
    return x+y  
  
a = f(1,2,f(1,2))  
  
print(a)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem D4. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def g(x,L=None):  
    if L is None:  
        L=[]  
    L.append(x)  
    return L  
  
def f(x, L=[]):  
    L = g(x)  
    return L  
  
L = f(2)  
L = f(3)  
print(L)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem D5. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
x = 2

def f():
    y = x + 1
    x = 4
    print(x+y)

f()
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem D6. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
x = 2

def f():
    x += 1
    print(x)

f()
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem D7. Below is a Python code snippet. First, identify if there are any errors. Then, provide a brief explanation for why each error occurs. If there is no error, write the output of the code.

```
def h(*K):
    return K[0] + K[1]

def f(**K):
    for item in K.values():
        print(item)

def g(**K):
    index1 = h(K["a"], K["b"] + K["c"], K["a"] + K["c"])
    index2 = h(K["b"] + K["c"], index1)
    f(a=index1, b=index2)

g(a=1, b=2, c=3)
```

If there are errors, specify the specific line(s) of code that are causing the error(s).

If there are errors, briefly explain why the error(s) occur(s):

If there is no error in the code, write the output of the code.

Problem E1. Briefly explain the role of the `pack()` functions in the following Python code. For instance, what happens if we do not call `E.pack()`?

```
import tkinter as tk

W = tk.Tk()
W.title("Python")

L = tk.Label(W, text = "Label")
L.pack()

def f():
    D.config(text = myText.get())

myText = tk.StringVar()
E = tk.Entry(W, textvariable = myText)
E.pack()

B = tk.Button(W, text = "Action", command = f)
B.pack()

D = tk.Label(W, text = "")
D.pack()

W.mainloop()
```

Your answer:

Problem E2. Write the output of the Python code.

```
import numpy as np
x = np.random.randint(5, size=(2, 3, 4))
print(x.size)
```

Your answer: