CS417 Programming Lab #1

This lab is meant to get you started using `ifs`, `for` loops, lists, and functions.

Before you begin

1. First, create a folder in your drive, and call it `cs417`. In that folder, create a sub-folder called `lab01`.
2. Then, download the starting program file `cond_loop.py` . Find folder `lab01`, and save it there.
3. Then, run `spyder`, or `IDLE`, for a python programming environment. and open the file `cond_loop.py` . Complete the tasks below.

Your tasks

1. Here is a python function that is passed two numbers, and prints them in sorted order:

```
def sort_two(a, b):
    if a < b:
        print(a, b)
    elif b < a:
        print(b, a)
    else:
        print(b, a)
```

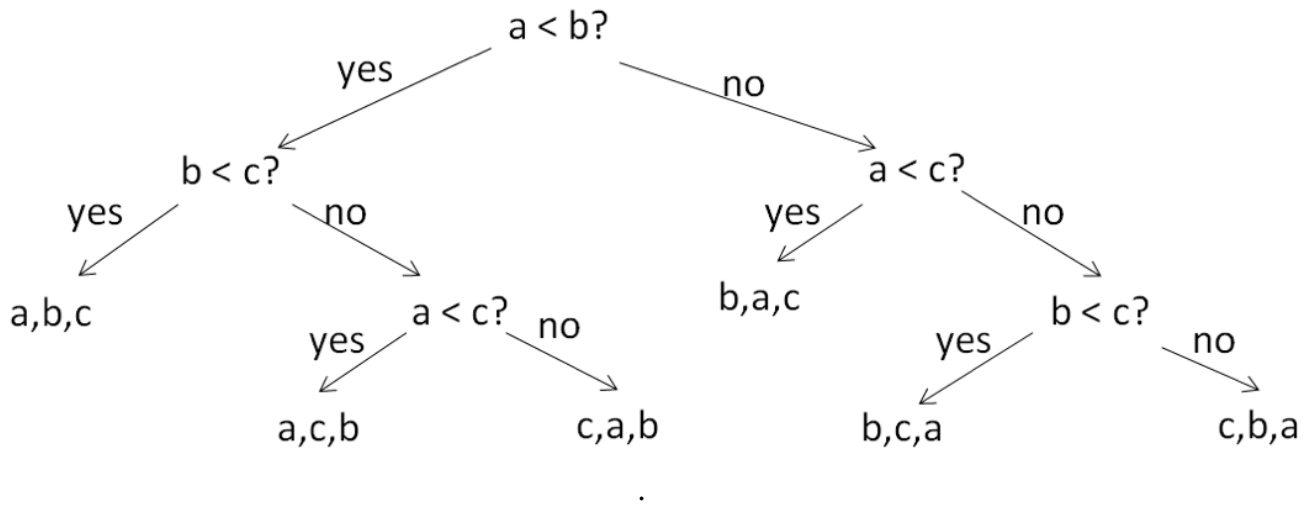
Notice the use of `if-elif-else`. This could have been written more simply:

```
def sort_two(a, b):
    if a < b:
        print(a, b)
    else:
        print(b, a)
```

Write a function `sort_three(a,b,c)` that prints three values in sorted order, for three numbers `a`, `b`, `c`. You *MAY NOT* put them in a list, and call the `sorted` function (that's too easy). Instead, you may ONLY use the following

```
if  else  elif  print()
```

To keep your thoughts organized, here is a decision tree for the problem:



.

---

2. Write a function, `delete_at(alist, k)` that takes a list, and an index `k`, and returns a new list, with the element at index `k` removed.

   Do not use the `pop()`, `insert()`, or `remove()` list methods. Instead, use list slices. For example, here we remove the element at index 2:

```
alist = [1, 2, 3, 4, 5, 6]
blist = alist[:2] + alist[3:]
```

---

3. Write a function, `insert_at(alist, i, x)` that takes a list, and an index `i`, and returns a new list, with the element `x` inserted at `i`. Do not use the `pop()`, `insert()`, or `remove()` list methods. Instead, use list slices. For example, here we insert 3 at index 2:

```
alist = [1, 2, 4, 5, 6]
blist = alist[:2] + [3] + alist[2:]
```

---

4. Write a function `max_difference(list1, list2)` that

- takes two lists of numbers,
- considers all numbers `num1` in the first list, and
- all numbers `num2` in the second list,
- and finds the biggest difference `abs(num1 - num2)`,
- and returns that difference.

For example, `max_difference([3, 2, 6, -5], [-4, 7, 3])` returns `12`, because -5 and 7 are farthest apart.

You will need a variable `biggest_difference` that tracks the biggest difference so far. Initialize it to 0, before looping.

You will need two nested `for`-loops, one indented inside the other (one loop for `list1` and a loop for `list2`).

Turning in your work

When you are done, click the “Submit Assignment” button, and upload `cond_loop.py`. You should turn in your work at the end of the lab session, even if you haven’t completed it. You will have until midnight to turn in your work again without a lateness penalty.

SAVE YOUR WORK!

The lab machines may not preserve your work. Next time, it may be gone. So, go to UNH BOX (google that), and upload your file `cond_loop.py`. This way, you can work from home, by going to BOX and downloading the file onto your computer.

Alternatively, you can save the file on a thumb drive, or email yourself the files.