Python for Math and Stat Fall 2023
Exam 2

Assume that all necessary packages have been imported.

1. (20 pts) For the following 4 problems, write down what each code block would display if executed in a Jupyter cell. If the code generates an error or infinite loop, write `Error`.

   (a)
   ```
   date = 'Nov 3 2023'
   date.split('2')
   ```

   (b)
   ```
   yr = 2023
   f'yr {yr/100:4.1f}'
   ```

   (c)
   ```
   num = 123456789
   result = []

   while num % 10 > 5:
       num //= 1000
       result.append(num)
   result
   ```

   (d)
   ```
   def func(alist):
       print(alist, end=' ')
       if len(alist) == 1:
           return 10

       return func(alist[1:]) + 10

   func([6, 8, 2])
   ```

   Solution:

   (a) `['Nov 3 ', '0', '3']`

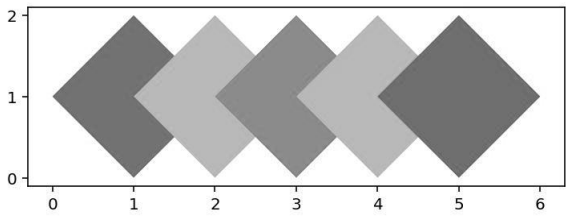
   (b) `'yr 20.2'`

   (c) `[123456, 123]`

   (d)
   ```
   [6, 8, 2] [8, 2] [2]
   30
   ```

2. (10 pts) Write a function **`diamonds(n)`** that uses `plt.fill()` to display a row of `n` overlapping diamonds. Each diamond measures 2 units wide and 2 units high. The diamonds overlap by one unit. Use default settings for color and aspect ratio.

   Example: `diamonds(5)` produces the following result.



   Solution:

```
def diamonds(n):
    for i in range(n):
        plt.fill([i, i+1, i+2, i+1], [1, 0, 1, 2])
    plt.show()
```

3. (10 pts) Suppose `randnums` is a list of positive integers. We wish to count the number of occurrences of each distinct integer. Write code to create a dictionary called **`freq`** with keys corresponding to the distinct integers in `randnums`. Each dictionary value equals the number of occurrences of its key in `randnums`.

   Example: If `randnums = [65, 999, 4, 65, 8, 4]`, then the final value of `freq` will be `{65: 2, 999: 1, 4: 2, 8: 1}` with the entries in some order.

   Note: The code should loop through `randnums`, incrementing the appropriate counter in `freq`. The code should not loop through the dictionary or use `count()`. It should create only the necessary key-value pairs.

   Solution:

   This solution initializes the dictionary with distinct keys and zero counts before tabulating.

```
freq = {i: 0 for i in set(randnums)}

for num in randnums:
    freq[num] += 1
```

   This solution creates an empty dictionary, then loops through `randnums`, inserting new keys when necessary.

```
freq = {}
for num in randnums:
    if num in freq:
        freq[num] += 1
    else:
        freq[num] = 1
```

4. (10 pts) For user accounts on the Buffers.com site, passwords must contain at least 6 characters with at least one letter and one digit. Write a function **`is_valid(passw)`** that returns `True` if a user's password `passw` satisfies the requirements. It returns `False` otherwise. Assume that `passw` is a string.

   Example: `is_valid('i8.cake')` returns `True` and `is_valid('I ate pie')` returns `False`.

   Hint: You may wish to use the `.isalpha()` and `.isnumeric()` methods which return `True` if a character is a letter or digit, respectively.

   Solution:

   This solution checks all the characters in `passw` before returning.

```
def is_valid(passw):
    if len(passw) < 6:
        return False

    letter = digit = False
    for ch in passw:
        if ch.isalpha():
            letter = True
        elif ch.isnumeric():
            digit = True

    return letter and digit
```

   This solution returns as soon as the password is confirmed to be valid.

```
def is_valid(passw):
    if len(passw) < 6:
        return False

    letter = digit = False
    for ch in passw:
        if ch.isalpha():
            letter = True
        elif ch.isnumeric():
            digit = True

        if letter and digit:
            break

    return letter and digit
```

   This solution uses a `while` loop and returns as soon as the password is confirmed to be valid.

```
def is_valid(passw):
    passw_len = len(passw)
    if passw_len < 6:
        return False

    letter = digit = False
```

```
i = 0
while not (letter and digit) and (i < passw_len):
    if passw[i].isalpha():
        letter = True
    elif passw[i].isnumeric():
        digit = True
    i += 1

return letter and digit
```