`Directions:` The exam is 120 minutes long. Please read each question carefully.

When asked to write code, you should write working Python code that has correct syntax. You should explain in 1-2 sentences what the idea for your solution is or write next to your code what it is doing. This will increase your chances of getting full/partial credit.
Use the backs of the pages if needed.

Last Name: _____

First Name: _____

Student ID #: _____

Question	Points	Score
1	20	
2	20	
3	20	
4	20	
5	20	
6	20	
7	20	
8	20	
Total:	160	

1. (20 points) Write down the output of the following programs.

1.
```
x = 1
s = 0
for i in range(8):
    s += x
    x += 1
print(s)
```

2.
```
def f(n):
    if n > 0:
        return n * g(n)
    return 1

def g(n):
    return f(n // 2)

print(f(6))
```

3.
```
from functools import reduce
x = reduce(lambda a,d: 2*a+d, [1,0,0,0,0,1,0,1])
print(x)
```

4.
```
def f(xs):
    if xs == []:
        return 0
    return xs[0] + f(xs[1:])

f([1,2,3,4,5])
```

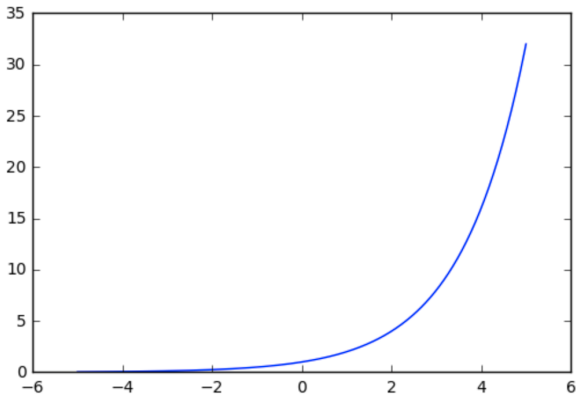
2. (20 points) Produce the following lists without using for or while loops.

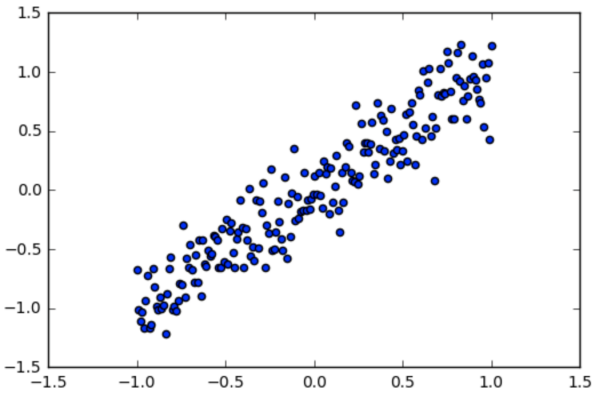
   1. `[0, 1, 3, 7, 15, 31, 63, 127, 255, 511]`

   2. `[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19]`

   3. `[-1, 2, -3, 4, -5, 6, -7, 8, -9, 10, -11, 12, -13, 14]`

3. (20 points) Write code that will produce the following graphs (or something that looks like it; use `plt.plot(X, Y)` and `plt.scatter(X, Y)`).

1. 

2. 

4. (20 points) Complete the code below to implement the function `chessboard(n)` that will return a numpy array with 1's and 0's arranged in a chessboard pattern. Assume `n` is odd. Examples:

```
In:   chessboard(3)
Out:  array([[0, 1, 0],
             [1, 0, 1],
             [0, 1, 0]])

In:   chessboard(5)
Out   array([[0, 1, 0, 1, 0],
             [1, 0, 1, 0, 1],
             [0, 1, 0, 1, 0],
             [1, 0, 1, 0, 1],
             [0, 1, 0, 1, 0]])
```

```
def chessboard(n):
    X =
    return
```

Complete the code below to implement the function `chessgonewrong(n)`, which produces a chess-board with the middle 3×3 square having -1's instead of 1s.

```
In:   chessgonewrong(7)
Out:  array([[ 0,  1,  0,  1,  0,  1,  0],
             [ 1,  0,  1,  0,  1,  0,  1],
             [ 0,  1,  0, -1,  0,  1,  0],
             [ 1,  0, -1,  0, -1,  0,  1],
             [ 0,  1,  0, -1,  0,  1,  0],
             [ 1,  0,  1,  0,  1,  0,  1],
             [ 0,  1,  0,  1,  0,  1,  0]])
```

```
def chessgonewrong(n):
    X = chessboard(n)

    return X
```

5. (20 points) Implement a function `divisors(n)` that returns all positive integer divisors of an integer `n` as a list. (returns not prints)

6. (20 points) A palindrome is a word that is the same when reversed, e.g. "amanaplanacanalpanama". Write a function `ispalin(s)` that will return `True` if a string `s` is a palindrome and `False` otherwise. (remark: you can work with `s` as if it were a list).

7. (20 points) Recall the `Polynomial` class from the homework that stores a polynomial as a list of its coefficients. Implement the `__add__(self, other)` function that returns a new polynomial which represents the sum of the polynomials `self` and `other`.

```
class Polynomial():
    def __init__(self, xs):
        self.coeffs = xs

    # returns a string representation of the polynomial
    def __repr__(self):
        if self.coeffs == []:
            return "0"
        c = ""
        for i, x in enumerate(self.coeffs):
            c += str(x) + "x" + "^" + str(i) + " + "
        return c[:-3]

    def __add__(self, other):
```

8. (20 points) Write code that will find the minimum of the function $f(x, y) = x^4 + y^2 + 2x + 4y + 1$ using gradient descent. (Start the descent from $(x, y) = (5, 5)$ and use the learning rate of $\eta = 0.01$). Your code should print the minimum value it finds.