1. The power set of S is { {ϕ}, {1}, {2}, {1, 2} } with ϕ being the empty set.

2. **Yes, they are equal:**

A = {1, 2}, B = {2, 1}	Given.
$(1 \epsilon A) \wedge (1 \epsilon B)$	1 exists in A and B
$(2 \epsilon A) \wedge (2 \epsilon B)$	2 exists in A and B
$(A \subseteq B) \wedge (B \subseteq A)$	A is a subset of B and B is a subset of A
A = B	Conclusion

3. ϕ and { ϕ } are not the same set. ϕ represents the empty set, while { ϕ } represents a set that's only element is the empty set.

4. Yes, the set (2, 3] is equivalent to $2 < x \leq 3$. (2, 3] is an infinite set that contains real numbers between 2 and 3. '(' defines the set to be non-inclusive to 2, and ']' defines the set to be inclusive of 3.

5. **$n \geq 4$**

6. $\bar{S} = U - S$ U is the universal set
   S = (2, 5)
   $(-\infty, 2] \cup [5, \infty)$

7. **Range of f is [0, 4].**
   $f(x) = x^2$
   the derivative of f(x), $f'(x) = 2x$
   In the domain [0, 2], $f'(x) \geq 0$.
   Therefore, f is an increasing function.
   f(0) = 0, and f(2) = 4

8. **No, $f(x) = x^2$ is not injective.**
   f(x) does not pass the horizontal line test:
   $f(1) = 1^2 = 1$
   $f(-1) = (-1)^2 = 1$
   f(1) = f(-1) therefore f(x) is not injective.

9. Show that f is both injective and surjective by definition of bijective:
   f(0, 0) = f(0, 0 XOR 0) = (0, 0)
   f(0, 1) = f(0, 0 XOR 1) = (0, 1)
   f(1, 0) = f(1, 1 XOR 0) = (1, 1)
   f(1, 1) = f(1, 1 XOR 1) = (1, 0)
   Because the f is one-to-one, it is injective.
   Because the domain of f is equal to its range, it is surjective.

Therefore, f is bijective.

Showing g is not bijective:
g(0, 0) = f(0, 0 AND 0) = (0, 0)
g(0, 1) = f(0, 0 AND 1) = (0, 0)
g(1, 0) = f(1, 1 AND 0) = (1, 0)
g(1, 1) = f(1, 1 AND 1) = (1, 1)
Because g is not one-to-one, it is not bijective.

Showing h is not bijective:
h(0, 0) = f(0, 0 OR 0) = (0, 0)
h(0, 1) = f(0, 0 OR 1) = (0, 1)
h(1, 0) = f(1, 1 OR 0) = (1, 1)
h(1, 1) = f(1, 1 OR 1) = (1, 1)
Because h is not one-to-one, it is not bijective.

As to the relation to the question in Homework 1, this is why the AND and OR operations could not be used to build a key (the C and D disks) for bit recovery. The lack of unique outputs (one-to-one) prevents the original bits from being determined by the key.

10. If $1 < 3x + 5 < 2$, then $\lceil 3x + 5 \rceil = 2$ Given by the ceiling function
If $2 < 3x + 5 \leq 3$, then $\lceil 3x + 5 \rceil = 3$. Given by the ceiling function
$1 < \lceil 3x + 5 \rceil \leq 3$ Subtract 5 from the inequality
$-4 < 3x \leq -2$ Divide inequality by 3
$\frac{-4}{3} < x \leq \frac{-2}{3}$ Solution

11. The formula from the lecture is: $\sum_{k=0}^{n} q^k = \frac{q^{n+1}-1}{q-1}$.

$\sum_{k=0}^{7} 6^k = \frac{6^8-1}{5}$ **is the simplification of the geometric sum**

12. $a_3 = 5$, $a_{11} = 87$ Given
$a_n = a_1 + d(n-1)$ Given formula
$a_3 = a_1 + d(2)$ Plug a_3 into formula
$a_{11} = a_1 + d(10)$ Plug a_{11} into formula
$a_{11} - a_3 = a_1 + d(10) - a_1 - d(2) = 87 - 5$ Subtract
$d(8) = 82$ Simplify
$\mathbf{d = \frac{82}{8}}$ Conclusion

13. $\sum_{k=2}^{1000} \frac{3^{2k+4}}{2^{3k+5}}$	Given
$\sum_{k=2}^{1000} \frac{(3^{2k})(3^4)}{(2^{3k})(2^5)}$	Simplify with rules of exponents
$\frac{3^4}{2^5} \sum_{k=2}^{1000} \frac{3^{2k}}{2^{3k}}$	Move constant to front of summation.
$\frac{81}{32} \sum_{k=2}^{1000} \left(\frac{9}{8}\right)^k$	Simplify and rewrite in the form: $\sum_{k=n}^{N} q^k$
$\left(\frac{81}{32}\right) [\frac{(9/8)^{1001} - (9/8)^2}{(9/8) - 1}]$	Use the given formula from the lecture: $\frac{q^{N+1} - q^n}{q-1}$
$\left(\frac{81}{32}\right) (8) \left(81/64\right) [(9/8)^{999} - 1]$	Simplify denominator, and factor out $\left(9/8\right)^2$
$25.6239 * \ [(9/8)^{999} - 1]$	Simplify and end.

14. EXTRA CREDIT:

```python
import math
# header
print("Name: William Jedynak")
print("ID  : 1227139214\n")
# lists to store data
fx = []
fn = []
n = 10
# initialize f(x)
for i in range(0, 21):
    if i == 5:
        fx.append(0)
    elif i == 12:
        fx.append(6)
    elif i == 20:
        fx.append(13)
    else:
        fx.append(i + 1)
# store values of pow(f(x), n)
for n in range(1, n + 1):
    for i in fx:
        fn.append(math.pow(i, n))
# create a new list and remove duplicates
lst = []
[lst.append(x) for x in fn if x not in lst]
# output relevant data
print("power value (n) is: " + str(n))
print("the count of all pow(f, n) is: " + str(len(fn)))
print("the count of distinct pow(f, n) is: " + str(len(lst)))
```

main

```
"C:\Users\WillJedynak\OneDrive\ASU Online\Fall 2022\Session B\MAT
Name: William Jedynak
ID  : 1227139214

power value (n) is: 10
the count of all pow(f, n) is: 210
the count of distinct pow(f, n) is: 171

Process finished with exit code 0
```