The Protolympics

Easy: (5 points each)

1. Take in an integer *n* and print out whether it is weird or not. A number is weird if it's odd, or if it is in the range of 6 and 20 (inclusive), other than that it isn't weird.

Input:	Output:
3	Weird
24	Not Weird

2. Take in a string *s* and output it a string with no consecutive repeated letters (no 2 letters are repeated on after the other).

Input:	Output:
soooosssooo	soso
heeeellllooo worrlddd	hello world

3. Check if a list is sorted or not. You can take in the numbers as a string and convert it to a list. How numbers are input doesn't matter.

Input:	Output:
1, 3, 2, 4, 5, 6	No
5, 6, 7	Yes

4. Ahmad likes numbers from ***A*** to ***B***, while Mahmoud likes numbers from ***C*** to ***D***. Is there any chance that there's a number they both like? Input format is ***A B C D***.

Input:	Output:
10 20 30 40	No
10 30 20 40	Yes

5. You'll take in ***n*** groups of numbers, represented as ranges (a number as the start and another as the end). Your job is to calculate the sum of the numbers between the ranges.

Input:	Output:
2	21
1 3	
4 6	

Explanation: First line of input contains ***n***, which is 2. Then 2 groups are entered, 1 to 3 and 4 to 6. 1st sum is 1+2+3 = 6 and 2nd sum is 4+5+6 = 15, and 6+15 = 21

6. A family consists of 3 people (mother, father, and son), and each of them uses ***x*** ml of shampoo each day. You'll take the size of the shampoo bottle, and then how many each family member uses each day. Your task is to figure out who'll run out of shampoo first.

Input:	Output:
30, 13, 11, 12	son
30, 10, 10, 10	mother

Case 1 Explanation: The shampoo bottle is the first number (30), the mother will use 13, leaving 17, then the father uses 11 leaving 6, and so the son can't shower.

Case 2 Explanation: The whole family will shower on the 1st day, but on the 2nd it will be finished so mother won't be able to shower.

7. Create a function that "zips" 2 lists (of the same size) together into 1 list made of tuples, where each tuple is made of an item from each list.

Input:	Output:
1, 2, 3, 4, 5	[(1,40),(2,50),(3,70),(4,90),(5,20)]
40, 50, 70, 90, 20	

Medium: (10 points each)

8. You'll take in a number from the Fibonacci sequence, your job is to return its index in the series.

Input:	Output:
55	10
13	8

9. Check if a sentence is a pangram or not. A pangram is a sentence that contains all of the letters of the alphabet at least once.

Input:	Output:
The quick brown fox jumps over a lazy dog	Yes
I love protons	No

10. You are given a string, and your job is to return the string without any duplicates.

Input:	Output:
ababacd	abcd
xxyaehheia	xyaehi

11. You'll take the ***time*** in the 24 hour format, and your job is to find out what the time will be after ***x*** minutes. Input format is ***time***, ***x***.

Input:	Output:
23:59, 10	after 10 mins it'll be 00:09
20:20, 121	after 121 mins it'll be 22:21

12. You'll take in a string of 1's and 0's. In one step you can eliminate "10" or "01" from the string, and you have unlimited steps. Your task is to print out the length of the string after removing all the "01" and "10"'s you can.

Input:	Output:	
1100	0	*(remove middle 10 then 10)*
01010	1	*(remove 1st 01 then 2nd 01)*
11101111	6	*(remove 10 or 01)*

13. You're given a list of numbers, your task is to find the largest sum of any increasing sub-array.

Input:	Output:	
10 20 30 5 10 50	65	*(sum of 5, 10, and 50)*
10 60 4 5 50	70	*(sum of 10 and 60)*

14. Given a ***string*** of length ***n***, you'll also take a number ***k*** which is a factor of ***n***. Your job is to divide the string into n/k substrings and eliminate the repetitions in each one. Input is ***string***, ***k***.

Input:	Output:
AAABCADDE, 3	ABCADE
ABBCCCDF, 4	ABCCDF

Hard: (20 points each)

15. Create a function that formats numbers so that between each 3 digits there is a comma (",").

Input:	Output:
100000	10,000
12345678	12,345,678

16. Ahmad and Mahmoud decided to play another game where when given a string, Ahmad creates substrings starting with consonants and Mahmoud creates substrings starting with vowels, and each counts their occurrences in the initial string. The score of each player is the sum of the occurrences of their group of substrings. Who'll win, and what is their score?

Input:	Output:
banana	Ahmad, 12

Explanation: Ahmad found the substrings ['b', 'ba', 'ban', 'bana', 'banan', 'banana', 'n', 'na', 'nan', 'nana'], which occurred 12 times, while Mahmoud found the substrings ['a', 'an', 'ana', 'anan', 'anana'] which occurred 8 times.

17. The sine of an angle ***x*** can be approximated by calculating the result of the first ***N*** terms of the series:

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \ (x \text{ in radians})$$

You'll take in the angle ***x*** and the number of terms ***N*** to use.

Bonus (5 points): Write a function that converts from degrees to radians so you can do the main problem more easily.

Input:	Output:
30, 100	0.5
40, 3	0.64
45, 10	0.71

18. You'll take in a number ***X***, and your job is to figure out how many numbers (A, B, and C) can satisfy the equation **"X = A * B + C"**.

Input:	Output:
3	3
100	473

Case 1 Explanation: Possible combinations are (1,1,2),(1,2,1),(2,1,1)

19. Given a string composed of letters and numbers, calculate the sum of all the numbers in the string.

Input:	Output:	
a12b34cd	46	*(12 + 34)*
a0bc123	123	*(0 + 123)*

20. Since you've mastered Python now, why not create a new language? Long ago, someone thought of the same idea and created a useless language called Braincrash (censored name). The language is composed of only 8 characters (+ - < > . , [ ]), and handles cells (maybe list items?). Your job is to create an interpreter for the language excluding the "[" and "]" characters.

Bonus (10 points): Implement "[" and "]" too.

Character	Meaning
`>`	Increment the data pointer (to point to the next cell to the right).
`<`	Decrement the data pointer (to point to the next cell to the left).
`+`	Increment (increase by one) the byte at the data pointer.
`-`	Decrement (decrease by one) the byte at the data pointer.
`.`	Output the byte at the data pointer.
`,`	Accept one byte of input, storing its value in the byte at the data pointer.
`[`	If the byte at the data pointer is zero, then instead of moving the instruction pointer forward to the next command, jump it *forward* to the command after the *matching* `]` command.
`]`	If the byte at the data pointer is nonzero, then instead of moving the instruction pointer forward to the next command, jump it *back* to the command after the *matching* `[` command.

Input:	Output:
,>,<++.>++.	if input is "f" and "g" it'll print "h" and "i"

Input:	Output:
+[----->+++<]>+.---.+++++++..+++.	hello