Problem A:

- Way Too Long Words
https://codeforces.com/problemset/problem/71/A

- If the word is greater than 10 characters print the first letter of the string, string.length( ) - 2, and then the last letter of the string

Problem B:

- There Are Two Types of Burgers
https://codeforces.com/problemset/problem/1207/A

- This is a traditional greedy problem. Find out which sandwich generates more money and sell as many of those as you can first. If you run out then sell the other type of sandwich.

Problem C:

- Telephone number
https://codeforces.com/problemset/problem/1167/A

- You need to find if there is an 8 anywhere in the string followed by at least 10 other characters. Iterate through the string. If you find an '8' and the n - i >= 11, then the answer is YES, otherwise it is impossible.

Problem D:

- Yet Another Tetris Problem
https://codeforces.com/problemset/problem/1324/A

- Since you can only place a 2x1 block on the field, notice that you will only be able to clear the whole field if there are only even columns or if there are only odd columns. If a field consists of one odd column and one even column you will always be left with a single block and it will not be possible.

Problem E:

- Divisibility Problem

https://codeforces.com/problemset/problem/1328/A

- There are three cases. If a is already divisible by b (a%b==0) then the answer is 0. If a is less than b, then the answer is b-a because you have to increase a until it equals b. If a is greater than b, then the answer is b - (a%b). This is because a%b tells you the remainder and you must subtract that from b because that tells you how many you must add to receive the next number that b evenly divides.

Problem F:

- Vasya and Lanterns

https://codeforces.com/problemset/problem/492/B

- Sort the array and calculate the distance between each lantern and divide it by two. This tells you the radius that you need between two lanterns to light up the whole area between them. Take the maximum of this value between all of the lanterns. After this, you must then account for the distance between the beginning of the street and the first lantern, and the distance from the lantern to the end of the street. So after taking the maximum value between all of the lanterns, do ans = max(ans, location of first lantern), and ans = max(ans, length of street - location of last lantern). For this problem you must use the double because using a float will not give you a precise enough answer.