Question1

Write a function that takes a list and a number as arguments. Add the number to the end of the list, then remove the first element of the list. The function should then return the updated list.

Examples

```
next_in_line([5, 6, 7, 8, 9], 1) ➞ [6, 7, 8, 9, 1]

next_in_line([7, 6, 3, 23, 17], 10) ➞ [6, 3, 23, 17, 10]

next_in_line([1, 10, 20, 42 ], 6) ➞ [10, 20, 42, 6]

next_in_line([], 6) ➞ "No list has been selected"
```

Question2

Create the function that takes a list of dictionaries and returns the sum of people's budgets.

Examples

```
get_budgets([
  { "name": "John", "age": 21, "budget": 23000 },
  { "name": "Steve",  "age": 32, "budget": 40000 },
  { "name": "Martin",  "age": 16, "budget": 2700 }
]) ➞ 65700

get_budgets([
  { "name": "John",  "age": 21, "budget": 29000 },
  { "name": "Steve",  "age": 32, "budget": 32000 },
  { "name": "Martin",  "age": 16, "budget": 1600 }
]) ➞ 62600
```

Question3

Create a function that takes a string and returns a string with its letters in alphabetical order.

Examples

```
alphabet_soup("hello") ➞ "ehllo"

alphabet_soup("edabit") ➞ "abdeit"

alphabet_soup("hacker") ➞ "acehkr"

alphabet_soup("geek") ➞ "eegk"

alphabet_soup("javascript") ➞ "aacijprstv"
```

Question4

Suppose that you invest $10,000 for 10 years at an interest rate of 6% compounded monthly. What will be the value of your investment at the end of the 10 year period?

Create a function that accepts the principal `p`, the term in years `t`, the interest rate `r`, and the number of compounding periods per year `n`. The function returns the value at the end of term rounded to the nearest cent.

For the example above:

```
compound_interest(10000, 10, 0.06, 12) ➞ 18193.97
```

Note that the interest rate is given as a decimal and `n=12` because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

Examples

```
compound_interest(100, 1, 0.05, 1) ➞ 105.0

compound_interest(3500, 15, 0.1, 4) ➞ 15399.26

compound_interest(100000, 20, 0.15, 365) ➞ 2007316.26
```

Question5

Write a function that takes a list of elements and returns only the integers.

Examples

```
return_only_integer([9, 2, "space", "car", "lion", 16]) ➞ [9, 2, 16]

return_only_integer(["hello", 81, "basketball", 123, "fox"]) ➞ [81, 123]

return_only_integer([10, "121", 56, 20, "car", 3, "lion"]) ➞ [10, 56, 20,
3]

return_only_integer(["String",  True,  3.3,  1]) ➞ [1]
```