

Send an email with the solutions as attachments (.c, .cpp or if you have .h and .hpp files) to the following email address: tofihomework+2024info3en@gmail.com

You shouldn't attach the compiled programs. If you feel like you're sending a lot of files (5+) you can put them in a zip, but you don't need to.

Set

This task is the first half of Homework 6. You can hand them in together or just homework 5. Together they're worth twice the points of course. If you want to do both of them, you should check out the page of homework 6, all the information from this page is also included there.

Write a Set class that can store a given amount of unique integers.

- When constructing a new set we should be able to set the maximum number of integers it can store.
- We should be able to add a new element through a method. If the given number is already in the set, then don't add it and return **false**. Otherwise add the new element and return **true**.
- Make a method that returns the length of the set, in other words how many elements it contains.
- The class should have a copy constructor.
- Write a method called **in** that checks whether the given integer is in the set or not. (It should return true/false.)
- We don't need to have any logic to deal with whether a new element fits in an existing set, we can assume that the sets will always be created large enough. (But if you have the time you can do it. A very easy method would be to double the size of the storage array when we run out of space and copy the existing content.)

Example main function for testing:

```
int main(void) {
    Set A = Set(10);
    A.add(23);
    Set B = A;
    Set C = A;
    C.add(12);
    Set D = A;
    cout << D.add(23) << endl;
    cout << A.length() << endl;
    cout << B.in(23) << endl;
    cout << C.in(12) << endl;
    cout << D.length() << endl;
    return 0;
}
```

Hints at the end of the page (if you have the time/perseverance, first try to solve it without the help, or just think about how you would solve it).

For the storage we can use a dynamically allocated array. Create this in the constructor. We should also store the amount of elements already in the set. Don't forget to write the destructor. Make sure to create a new array in the copy constructor, you shouldn't copy the array's pointer.

One thing to keep in mind is to always correctly increment and decrement the variable that stores the current amount of elements.