

Informatics3-2024/Homework3

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.

Unique

Write a C++ function that takes an **int** array and creates a copy that contains only unique elements. In other words only the first instance of every number is present in the new array. This new array should then be returned.

The head of the function could look something like this (you don't have to use these variable names):

```
int* unique(int t[], int n, int *m);
```

The input array is **t** its length is **n** and the output array's length should be returned through the **m** pointer. (just like in the first homework).

As an example for the input array {1, 5, 4, 2, 2, 4, 3, 6, 2, 7} the output would be the {1, 5, 4, 2, 3, 6, 7} array.

Don't be afraid to use lots of **couts** in the code while you're testing or are looking for errors. It helps a lot if you can see what actually happens in there.

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).

The first difficulty you'll face is to figure out the length of the new array. We should return an array with the right size this time. An idea here would be to use a temporary array that's the exact size as the input array. We would work in this temporary array. At the end we'll copy this to a new array with the right length. Then we can free this temporary array and return the new, right length array.

Other than the length the algorithm isn't that difficult. You'll need a cycle to go through the input array and we'll write the values in the new (temporary) array. At every point we check whether the new value is already in the temporary array or not. To do this test you can create a new function that does it, or you can also do it in place with another cycle. As with the 2nd homework, you'll probably need to use 2 cycle variables. One for the input array and 1 for the temporary array (because once you skip an element the indices of the arrays misalign).