Design a **C++ class** that manages a **dynamic integer array** with the following requirements:

1. The class should allocate memory dynamically for an integer array.
2. Implement a **copy constructor** to ensure a **deep copy** of the dynamically allocated array.
3. Provide a **method to modify the array size** and update its values.
4. Implement a **print method** to display array elements.
5. Demonstrate the class functionality in the `main()` function by:
    - Creating an object with an initial size of 5 and input values.
    - Creating another object using the **copy constructor**.
    - Printing both objects.
    - Modifying the original object's size and values.
    - Printing both objects again to verify the deep copy mechanism.

Constraints:

- Ensure there is no memory leak when modifying the array size.
- Copy constructor should allocate a new array and copy the values from the existing object.

==

```
#include <iostream>
#include<cstring>
using namespace std;
class strOperations
{
  int* array;
  int size;
  public:
  strOperations(int s) : size(s)
  {
    array=new int[size];
    cout<<"Enter the "<<size<<"values : "<<endl;
    for(int i=0;i<size;i++)
    {
      cin>>array[i];
    }
  }
  strOperations(strOperations& obj): size(obj.size)
  {
    array=new int[size];
    for(int i=0;i<size;i++)
    {
            array[i]=obj.array[i];
        }

    }
    void set()
    {
        cout<<"Enter the new size"<<endl;
        cin>>size;
        array=new int[size];
        cout<<"Enter the "<<size<<" elements :"<<endl;
        for(int i=0;i<size;i++)
        cin>>array[i];
    }
    void print()
    {
        cout<<"The elements are : "<<endl;
        for(int i=0;i<size;i++)
        {
            cout<<array[i]<<endl;
        }
    }
    ~strOperations()
    {
        if(array!=nullptr)
        {
            delete[] array;
            array=nullptr;
        }
        cout<<"Memory de allocated"<<endl;
    }
};
int main()
{
    strOperations st(5);
    strOperations st1(st);
    st.print();
    st1.print();
    st.set();
    st.print();
    st1.print();
    return 0;
}
```