Objective:

- Issue related to Object transition by value: Copy Constructor
- Character Array manipulation.

Task-0:

Add copy constructor feature in your Array class.

Task-1:

Add/Update/Modify the following in your String class.

Note: You are not allowed to use any library functions related to strings.

```
class String
{
    char * data;
    int size;
public:
```

`String ( const String & );`	
`String ();`	Initializes data to nullptr and size to 0.
`String (char c);`	Initializes data with char c
`String(const char *);`	Initializes the data with received string by allocating memory on heap.
`~String ();`	You know what to do.
`void input();`	Takes input from console in *this object.
`char & at(int index);`	Index: Receives the index for string. Return Value: reference of array location represented by index
`bool isEmpty( );`	Tells whether string is empty or not? An empty string is one which has data==nullptr or data[0]=='\0'. Return Value: return true if string empty otherwise false.
`int getLenght();`	Returns length of the string.
`int getSize();`	Returns the size of array.
`void display();`	Prints the string on console.
`int find(String subStr, int start=0 ) ;`	Find the substring in the *this object. By default, search starts from 0 index. Returns the index at which subStr is found otherwise return -1.
`void insert(int index, String subStr);`	Insert the substring at given index in calling object.
`void remove(int index, int count=1);`	Remove the characters (how many? Given in count) starting from index
`int replace(String old, String newSubStr );`	Find all the occurrences of old substring and replace it with new substring. Return the count of occurrences found in calling object.
`void trimLeft();`	Removes all the white space characters on the left of string
`void trimRight();`	Removes all the white space characters on the right of string
`void trim();`	Removes all the white space characters on both left and right sides of string
`void makeUpper();`	Change all the alphabets to uppercase
`void makeLower();`	Change all the alphabets to lowercase
`void reverse();`	It reverses the string stored in the calling object
`void reSize(int);`	You know what to do. If receiving capacity <=0 then make the string size = 0 and deal data pointer accordingly.
`void shrink();`	Resize/shrink the array equal to the length of string pointed by data.
`int compare (String s2 );`	Compare the calling and receive object string. It should behave just like strcmp
`String left ( int count );`	Count: The number of characters to

	extract from calling object from left side Return Value: A String object that contains a copy of the specified range of characters
`String right ( int count );`	
`long long int convertToInteger ();`	Converts the integral value stored in calling object to long long int and returns the integral value.
`String concatenate (String s2 );`	It returns the concatenated result of received and calling object without changing calling object.

};

Task-2:

Explore the `strtok` function related to char arrays?
Following links are recommended for the said purpose.

https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm
https://www.cplusplus.com/reference/cstring/strtok/