Prototype	`int indexOf(char *str, char *substr)`
Description	Returns the index of the first occurrence of *substr* in *str*.
Parameters	**str** - the main string which will be checked **substr** - the substring which will be looked for in the main string
Returns	The index of the first occurrence of *substr* in *str* or -1 if *substr* is not in *str*.

Prototype	`char *getString (void)`
Description	Capture input from standard input as string.
Parameters	**Void**
Returns	The address of the string captured from the standard input.

Prototype	`int endsWith (char *str, char *substr)`
Description	Check either *substr* is located exactly at the end of *str* or not.
Parameters	**str** - the main string which will be checked **substr** - the substring which will be looked for in the main string
Returns	Returns ONE (true) if *substr* is located at the end of *str* or ZERO (false) if *substr* is not located at the end of *str*.

Prototype	`int equals (char *str1, char *str2)`
Description	Check whether *str1* and *str2* are the same or not. (Case sensitive)
Parameters	**str1** - the first string **str2** - the second string
Returns	Returns ONE (true) if *str1* and *str2* are the same or ZERO (false) if *str1* and *str2* are different.

Prototype	`int equalsIgnoreCase (char *str1, char *str2)`
Description	Check whether *str1* and *str2* are the same or not. (Case insensitive)
Parameters	**str1** - the first string **str2** - the second string
Returns	Returns ONE (true) if *str1* and *str2* are the same or ZERO (false) if *str1* and *str2* are different.

Prototype	`int length (char *str)`
Description	Returns the length of *str*.
Parameters	**str** - the string which will be counted.
Returns	Returns the length of *str*.

Prototype	`char *reverse (char *str)`
Description	Returns new copy *str* which is in reverse order.
Parameters	**str** - the string which will be reversed
Returns	The address of the copy of *str* which has been reversed.

Prototype	`char **split (char *str, char *delimiter)`
Description	Returns a set of strings which have been split from *str* by *delimiter*.
Parameters	**str** - the main string which will be splitted **delimiter** - the delimiter which marks the point for the string to be splitted.
Returns	The address of address of the string which has been split.

Prototype	`int startsWith (char *str, char *substr)`
Description	Check either *substr* is located exactly at the beginning of *str* or not.
Parameters	**str** - the main string which will be checked **substr** - the substring which will be looked for in the main string
Returns	Returns ONE (true) if *substr* is located at the beginning of *str* or ZERO (false) if *substr* is not located at the beginning of *str*.

Prototype	`char *toLowerCase (char *str)`
Description	Convert *str* into lower case.
Parameters	**str** - the string which will be converted
Returns	The address of the copy of *str* which has been converted into lower case.

Prototype	`char *toUpperCase (char *str)`
Description	Convert *str* into upper case.
Parameters	**str** - the string which will be converted
Returns	The address of the copy of *str* which has been converted into upper case.