Mandatory tasks

1. Write a program that prompts the user for a text up to 20 characters long and makes a copy of it. Make sure that the copy uses only the amount of memory that is absolutely necessary.

2. Create a function called reverse () that has a string parameter. The function translates the characters of the string locally. That is, the function returns the result in the same memory area that it received as a parameter. For example, the characters in a palindrome string do not change in the function.

3. Modify the previous function named reverse () so that you do not replace the characters locally, but create another string and return with it. Allocate the required memory dynamically

4. Write a program that reads 5 words from the user and prints them in reverse order. Use an array.

5. Write a program that waits for a number as a command line argument and reads as many words from the user as you specified. When you have entered all of them, write the words in reverse order. Use an array.

6. Modify the solution to the previous task so that if the user enters the word "END", the scan ends before reaching the limit specified as a command line argument. Use an array. Then write the contents of the array in reverse order.

7. Modify the solution of the previous problem so that the program no longer prompts the user for a command line argument, but reads words from the user until the input is EOF. Then print them out in reverse order. When you're done typing, wait for words again. Use an array. Be careful not to create memory debris.

Optional tasks

1. Create a function that expects an array of integers as parameters. The function should return with another array in which the elements of the parameter array are sorted: even at the front and odd at the back. Manage memory dynamically

Advanced tasks

1. Given a file with numbers separated by spaces in each line. Write a program that reads line by line from this file and prints each line in ascending numerical order to a file named output.txt. Manage memory dynamically