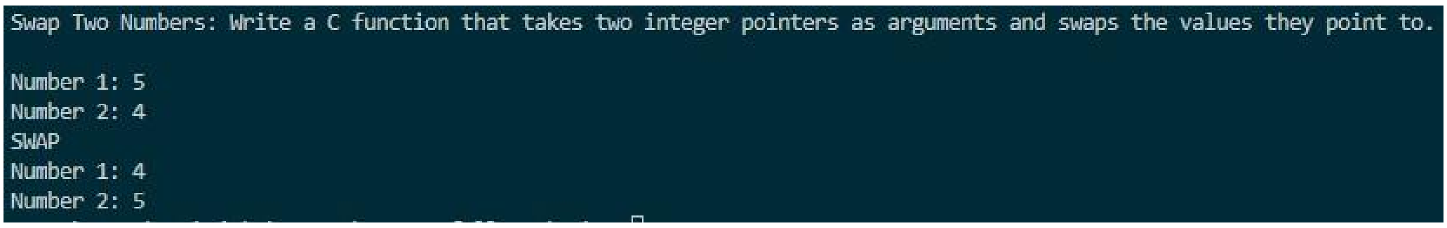
Dr. Chatterjee’s CS 3150
Report by Weston Shakespear
Fall 2023

# *Lab 3*

## 1. Swap Two Numbers:

Write a C function that takes two integer pointers as arguments and

swaps the values they point to.

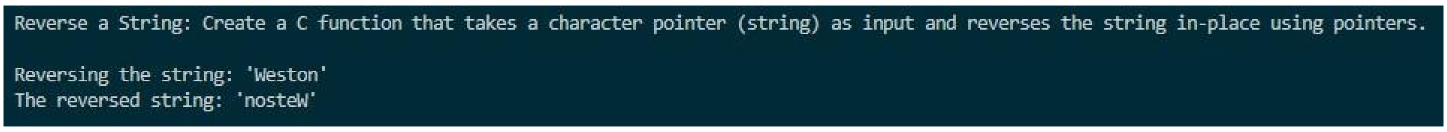


```
Swap Two Numbers: Write a C function that takes two integer pointers as arguments and swaps the values they point to.

Number 1: 5
Number 2: 4
SWAP
Number 1: 4
Number 2: 5
```

## 2. Reverse a String:

Create a C function that takes a character pointer (string) as input and

reverses the string in-place using pointers.

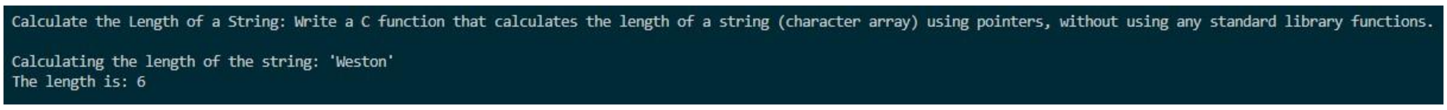


```
Reverse a String: Create a C function that takes a character pointer (string) as input and reverses the string in-place using pointers.

Reversing the string: 'Weston'
The reversed string: 'notseW'
```

## 3. Calculate the Length of a String:

Write a C function that calculates the length of a string

(character array) using pointers, without using any standard library functions.

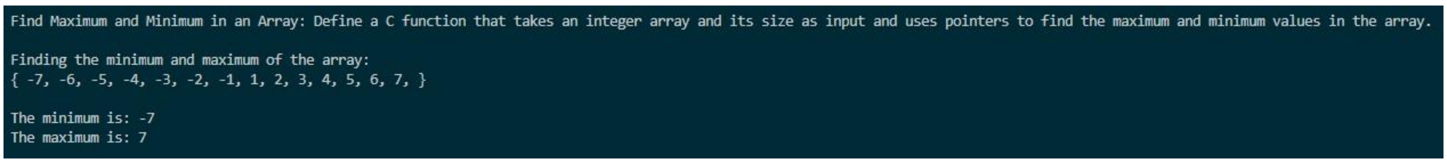


```
Calculate the Length of a String: Write a C function that calculates the length of a string (character array) using pointers, without using any standard library functions.

Calculating the length of the string: 'Weston'
The length is: 6
```

## 4. Find Maximum and Minimum in an Array:

Define a C function that takes an integer array

and its size as input and uses pointers to find the maximum and minimum values in the array.

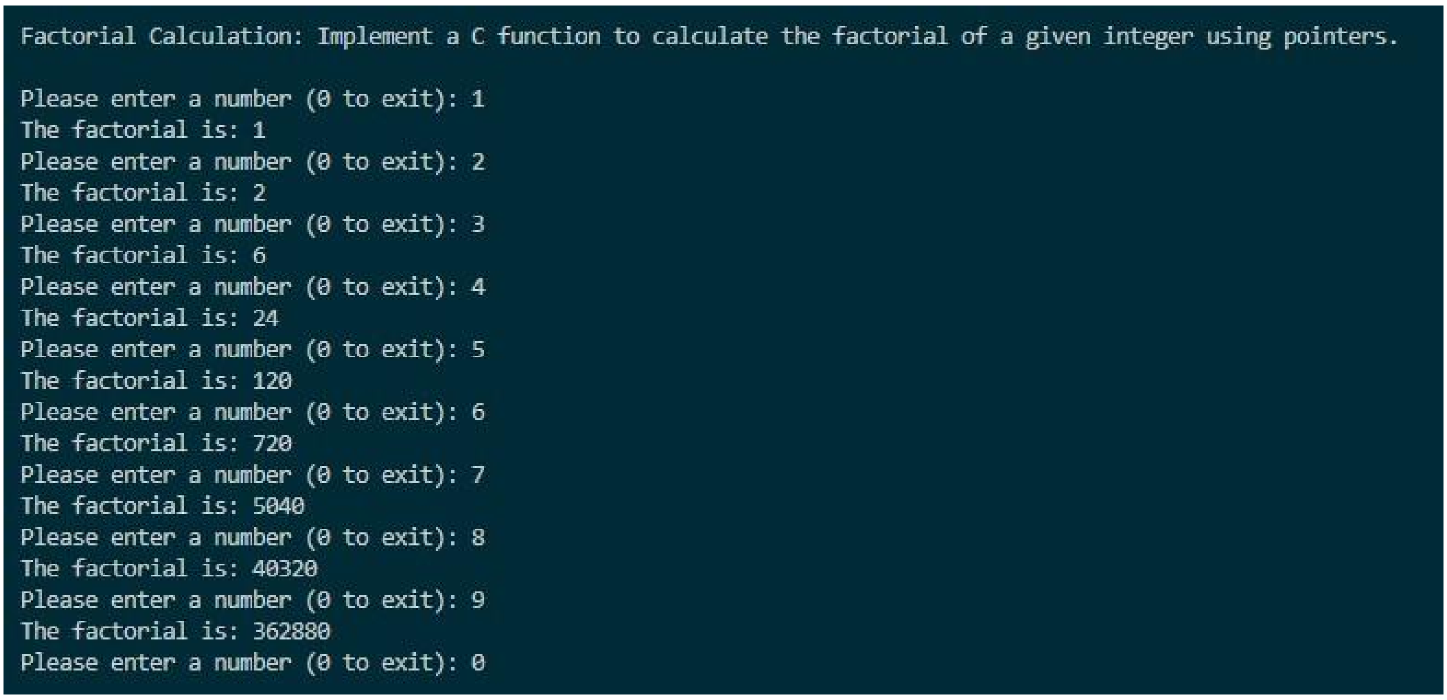


```
Find Maximum and Minimum in an Array: Define a C function that takes an integer array and its size as input and uses pointers to find the maximum and minimum values in the array.

Finding the minimum and maximum of the array:
{ -7, -6, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, }

The minimum is: -7
The maximum is: 7
```

## 5. Factorial Calculation:

Implement a C function to calculate the factorial of a given integer

using pointers.

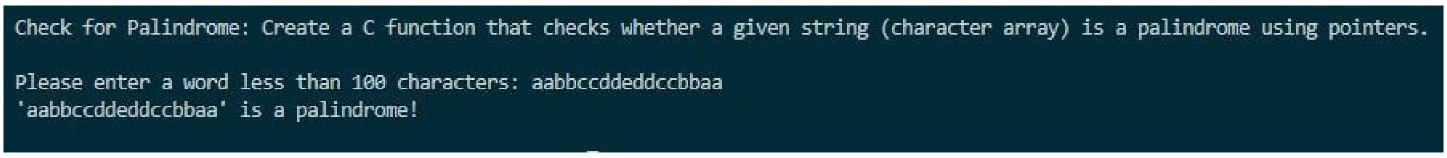


```
Factorial Calculation: Implement a C function to calculate the factorial of a given integer using pointers.

Please enter a number (0 to exit): 1
The factorial is: 1
Please enter a number (0 to exit): 2
The factorial is: 2
Please enter a number (0 to exit): 3
The factorial is: 6
Please enter a number (0 to exit): 4
The factorial is: 24
Please enter a number (0 to exit): 5
The factorial is: 120
Please enter a number (0 to exit): 6
The factorial is: 720
Please enter a number (0 to exit): 7
The factorial is: 5040
Please enter a number (0 to exit): 8
The factorial is: 40320
Please enter a number (0 to exit): 9
The factorial is: 362880
Please enter a number (0 to exit): 0
```

*Lab 3*

## 6. Check for Palindrome:

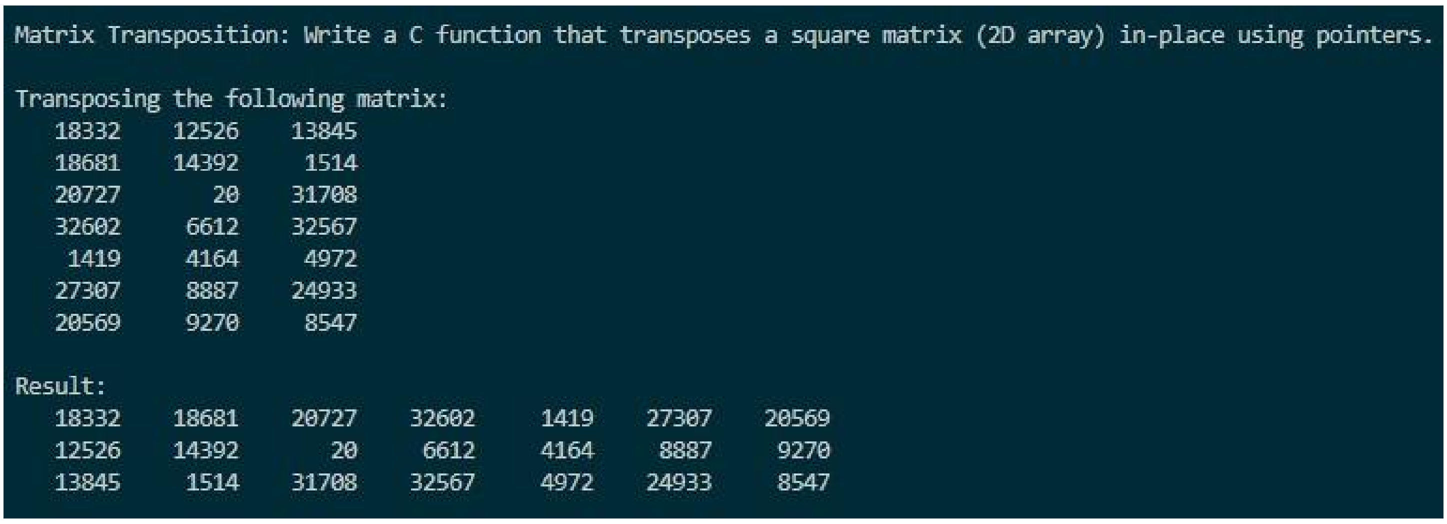
Create a C function that checks whether a given string (character array) is a palindrome using pointers.



```
Check for Palindrome: Create a C function that checks whether a given string (character array) is a palindrome using pointers.

Please enter a word less than 100 characters: aabbccddeddccbbaa
'aabbccddeddccbbaa' is a palindrome!
```

## 7. Matrix Transposition:

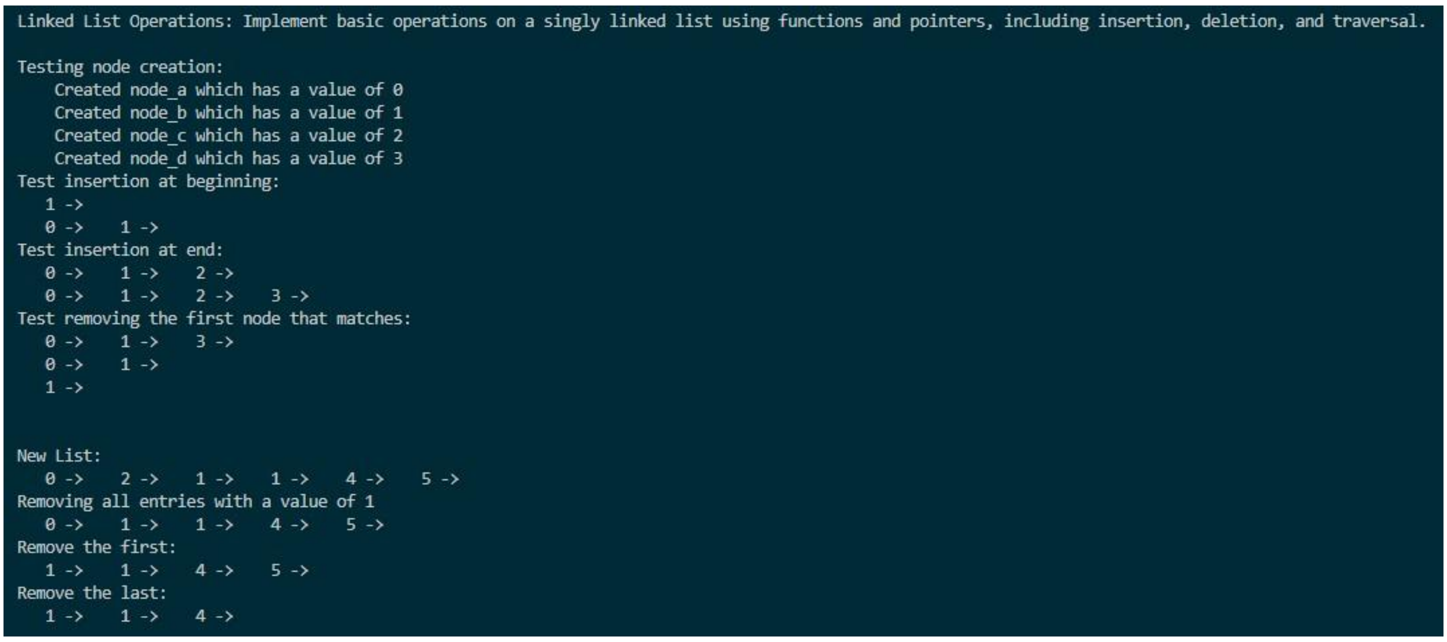
Write a C function that transposes a square matrix (2D array) in-place using pointers.



```
Matrix Transposition: Write a C function that transposes a square matrix (2D array) in-place using pointers.

Transposing the following matrix:
   18332    12526    13845
   18681    14392     1514
   20727       20    31708
   32602     6612    32567
    1419     4164     4972
   27307     8887    24933
   20569     9270     8547

Result:
   18332    18681    20727    32602     1419    27307    20569
   12526    14392       20     6612     4164     8887     9270
   13845     1514    31708    32567     4972    24933     8547
```

## 8. Linked List Operations:

Implement basic operations on a singly linked list using functions and pointers, including insertion, deletion, and traversal.



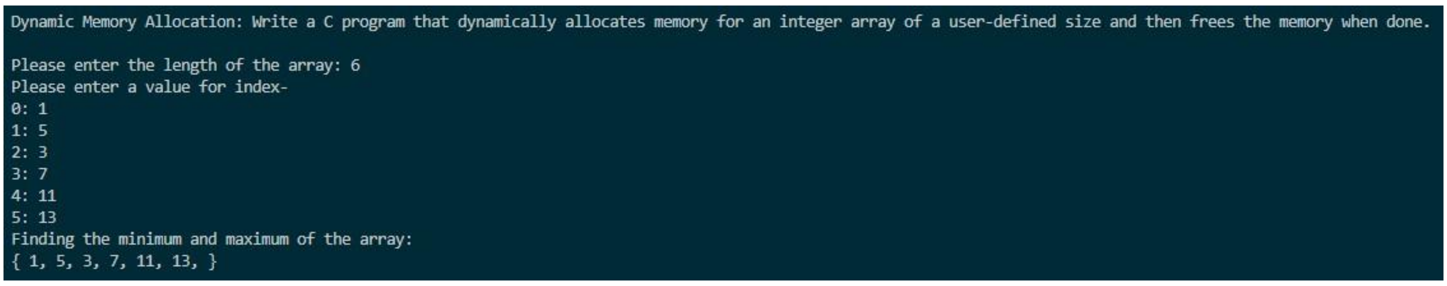
```
Linked List Operations: Implement basic operations on a singly linked list using functions and pointers, including insertion, deletion, and traversal.

Testing node creation:
    Created node_a which has a value of 0
    Created node_b which has a value of 1
    Created node_c which has a value of 2
    Created node_d which has a value of 3
Test insertion at beginning:
   1 ->
   0 ->    1 ->
Test insertion at end:
   0 ->    1 ->    2 ->
   0 ->    1 ->    2 ->    3 ->
Test removing the first node that matches:
   0 ->    1 ->    3 ->
   0 ->    1 ->
   1 ->


New List:
   0 ->    2 ->    1 ->    1 ->    4 ->    5 ->
Removing all entries with a value of 1
   0 ->    1 ->    1 ->    4 ->    5 ->
Remove the first:
   1 ->    1 ->    4 ->    5 ->
Remove the last:
   1 ->    1 ->    4 ->
```

# *Lab 3*

## 9. Dynamic Memory Allocation:

Write a C program that dynamically allocates memory for an

integer array of a user-defined size and then frees the memory when done.

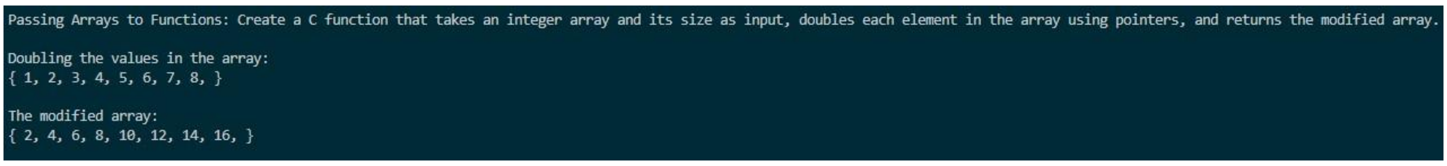


```
Dynamic Memory Allocation: Write a C program that dynamically allocates memory for an integer array of a user-defined size and then frees the memory when done.

Please enter the length of the array: 6
Please enter a value for index-
0: 1
1: 5
2: 3
3: 7
4: 11
5: 13
Finding the minimum and maximum of the array:
{ 1, 5, 3, 7, 11, 13, }
```

## 10. Passing Arrays to Functions:

Create a C function that takes an integer array and its size as

input, doubles each element in the array using pointers, and returns the modified array.



```
Passing Arrays to Functions: Create a C function that takes an integer array and its size as input, doubles each element in the array using pointers, and returns the modified array.

Doubling the values in the array:
{ 1, 2, 3, 4, 5, 6, 7, 8, }

The modified array:
{ 2, 4, 6, 8, 10, 12, 14, 16, }
```