

# CS 332/532 - Systems Programming

## HW 1

### Objectives

1. Practice C Programming

### Assignment Submission

The assignment submission in Canvas is mandatory. No late submissions will be accepted.

#### Submission Checklist:

- Upload a C source file (.c file) to Canvas as part of this assignment submission. Submissions through the Canvas “Comments” will not be accepted. The file should be named in this naming convention: yourblazerid\_HW01.c
- Upload a README.md file which should include:
  - Instructions on how to compile your C source file into an executable.
  - How to run the executable program.
  - Any citation documentation.
  - A link to your GitHub repository.
- Upload an Independent Completion Form.

Please do not upload executables or object files.

### Rubric

Criteria	Pts
Uploaded Independent Completion Form	5 pts
Uploaded .c file Uploaded .c file of code submission.	5 pts
Uploaded .md readme file Uploaded .md file of code description, usage information, and any coding citations.	5 pts
Provided working (shared) Github repository link Provided working (shared) Github repository link in the README file.	5 pts

Criteria	Pts
numberTransformer Compiles & Does Not Error	20 pts
UABNumber Compiles & Does Not Error	15 pts
reverseNum Compiles & Does Not Error	15 pts
smallerThanIndex Compiles & Does Not Error	15 pts
arrayDetails Compiles & Does Not Error	15 pts

Your code needs to be compiled on GitHub codespace, or you need to demo your code to TA.

## HW Assignment #1

Write a C program that is capable of the following functionality:

### 1. numberTransformer (n)

Write the function **numberTransformer (n)** that takes a positive integer 'n' and prints a string according to the following conditions:

- If 'n' is a prime number other than 3 or 5, the function should return: "Go Blazers".
- If 'n' is a power of 2 (e.g., 1, 2, 4, 8, ...), the function should return the result of adding 'n' and the nearest prime number below 'n' (as a string).
- If the sum of the digits of 'n' is divisible by 3, the function should return: "CS".
- If 'n' is divisible by both 3 and 5, the function should return: "UAB CS 332&532".
- Otherwise, the function should return the square of the sum of the digits of 'n' (as a string).

Sample Inputs	Expected Outputs
n=3	"CS"
n=64	"125" [hint: 64+61]
n=17	"Go Blazers"
n = 10	"1" [hint: (1+0) ^2]
n=30	"UAB CS 332&532"

## 2. UABNumber()

Write the function **UABNumber ()** that will ask the user to enter an integer and assign this value to an integer variable 'n2'. Your function will return a Boolean value (True or False). You will consider a number is a UABNumber if the value of the number is equal to the sum of its positive divisors. Your function will consider the input value and return True if the input parameter is a UABNumber and return False otherwise. While finding the positive divisors, do not include the number itself and assume 'n2' is equal or greater than 0.

Sample Inputs	Expected Outputs
n2=28	True *Hints=> 28=1+2+4+7+14
n2=12	False *Hints => 12 != 1+2+3+4+6
n2=6	True *Hints=> 6 = 1+2+3
n2=27	False *Hints => 27 != 1+3+9

## 3. reverseNum(n3)

Write the function **reverseNum (n3)** that takes an integer 'n3' and returns another integer. The function will reverse the order of the digits and return the new value. Assume the input will contain only positive integers.

Sample Input	Expected Output
n3 = 1234	4321
n3 = 29	92
n3 = 10001	10001

## 4. smallerThanIndex()

Write the function **smallerThanIndex( )** that takes an array of 5 integers (numbers) and returns an integer. The function will check every number's value and their indices. Count the number of integers in the array whose value is smaller than index and return the total.

Sample Input	Expected Output
numbers=[10,20,1,2,30]	2 *Hint (1 and 2)
numbers=[1,2,0,44,29]	1 *Hint (only 0)
numbers=[-4,-3,2,1,0]	4 *Hint (-4,-3,1,0)

## 5. **arrayDetails ()**

Write the function **arrayDetails ()** that takes in an array of 6 integers and prints another array containing (in the following order):

the minimum value, the minimum value's index, the mean (rounded to the nearest hundredth), the maximum value, and the maximum value's index (total of five elements).

Assume that the input will always be an array of 6 integers. Built-in methods and functions are permitted.

### **Sample input**

```
arr = [-8, 18, 103, 1, -4, 631]
```

### **Sample Output**

```
[-8, 0, 123.33, 631, 5]
```