

CS 332/532 – 1G- Systems Programming

HW 1

Deadline: 01/29/2023 Sunday 11:59pm

Objectives

Practice C Programming

Grading

- Submit the `blazerid_HW01.c` file to the Canvas.
- Each question is 20 points. A problem is correct if all our tests pass, otherwise it will be considered as False.
- The deadline is 01/29/2023 Sunday 11:59pm (no late submission will be accepted)
- This is **not** a group project, violation of the Academic Integrity Code will result in a failure grade.
- Your code needs to be compiled on Vulcan server, or you need to demo your code to TA
- ***** Do not forget to include "independent completion form" *****

`intro332532 (n)`

Write the function `intro332532` that takes a positive integer `n` and prints a string according to the following conditions

- if `n` is divisible by 5, it should print "UAB"
- if `n` is divisible by 3, it should print "CS"
- if `n` is divisible by both 3 and 5, it should print "UAB CS 332&532"
- if `n` is a prime number other than 3 or 5 it should print "Go Blazers"
- otherwise, it should print the cube of `n`

Sample Inputs	Expected Outputs
<code>n=3</code>	"CS"
<code>n=70</code>	"UAB"
<code>n=4</code>	64
<code>n=17</code>	"Go Blazers"
<code>n=30</code>	"UAB CS 332&532"

UABNumber ()

Write the function UABNumber that will ask user to enter an integer and assign this value to a 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, it will return False otherwise. While finding the positive divisors, do not include the number itself and assume n 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

reverseNum(n3)

Write the function “**reverseNum**” 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 the positive integers only

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

smallerThanIndex()

Write the function `smallerThanIndex()` that takes an array of integers (`numbers`) and return 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.

+10 Bonus points: Use pointers to manipulate the array

Sample Input:	Expected Output
<code>numbers=[10,20,1,2,30]</code>	2 <i>*Hint (1 and 2)</i>
<code>numbers=[1,2,0,44,29,309]</code>	1 <i>*Hint (only 0)</i>
<code>numbers=[-4,-3,2,1,0]</code>	4 <i>*Hint (-4,-3,1,0)</i>

arrayDetails()

Write a function **arrayDetails** that takes in an integer array **arr** and returns another array containing (in the following order) the *number of elements in the array*, 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 **six** elements). Assume that the input will always be an array of integers. **Built-in-methods and functions are permitted.**

Sample input:

`arr = [-8, -23, 18, 103, 0, 1, -4, 631, 3, -41, 5]`

Sample Output:

`[11, -41, 9, 62.27, 631, 7]`