

**GTU Department of Computer Engineering**  
**CSE 331 - Spring 2022**  
**Homework 1 Report**

**Emirkan Burak Yılmaz**  
**1901042659**

## 1. PROBLEM SOLUTION APPROACH

First, the problem inputs which are the length of the array  $n$ , and the integer divisor  $k$  are taken from the user as two space separated integers. Then the  $n$  space-separated integers are taken from the user and put in array  $ar$ . These user inputs are taken from the user as whole line string and then they are parsed and converted to integer. The problem constrains

$$2 \leq n \leq 100$$

$$1 \leq k \leq 100$$

$$1 \leq ar[i] \leq 100$$

are also checked after parsing string and converting the integer. For the cases, invalid inputs such as characters or improper values like -12 are detected and force user to enter valid inputs.

String conversion is done with the `stoi` function which takes the start address of an array, array length and the address of the string that is being converted in respectively registers `$a0`, `$a1`, `$a2` as arguments. Function `stoi` generates the integer value by examining the string byte by byte till the end of the string or the array is totally filled. When examination encounters a space character, it saves the generated integer value to the relevant array location.

After the inputs are converted and saved in registers then the function `divisible_sum_pairs` takes the flow of control and solves the problem.

Function `divisible_sum_pairs` takes start address of array, array length and the divider in respectively registers `$a0`, `$a1`, `$a1` as arguments. Firstly, the end address of the array is calculated by using the start address of array and its length. It also uses a counter to keep track of the number of divisible pairs. Then these addresses are used to keep track of the two nested loops. The outer loop executes from the start address to end address and the inner loop executes from the next position of the current address to end address in each loop. During the inner loop there are some calculations done to find the divisible pairs.

To find pairs the address values of outer loop and inner loop are used to load the integer values from memory to register. After having these two values they are summed up and divided with the divisor to check if it's divisible or not. After division, if the value of `$hi` register is 0 which means the sum is divisible by the divisor, the pair is printed, and pair counter increased by one. After the nested loop execution is done the counter value is saved in register `$v0` as return value.

## 2. TESTS

Mars Messages	Run I/O
Clear	Enter space-seperated integers array length n and divisor k: 5 dflgmkgfdg
	Invalid syntax. Please make sure to enter space between positive integers
	Enter space-seperated integers array length n and divisor k: -12 4
	Invalid syntax. Please make sure to enter space between positive integers
	Enter space-seperated integers array length n and divisor k: sdfdfs fsdf dfsdfs fsdf
	Invalid syntax. Please make sure to enter space between positive integers
	Enter space-seperated integers array length n and divisor k: 3 5
	Length (n): 3, Divisor (k): 5
	Enter 3 space-seperated integers: 4 6 35
	ar = [ 4 6 35 ]
(4, 6)	
Number of pair(s): 1	
-- program is finished running --	

Mars Messages	Run I/O
Clear	Length (n): 6, Divisor (k): 5
	Enter 6 space-seperated integers: 1 2 132 4 -3
	Each integer value v should be in the boundry $1 \leq v \leq 100$
	Enter 6 space-seperated integers: 1 2 101 fd sdf
	Each integer value v should be in the boundry $1 \leq v \leq 100$
	Enter 6 space-seperated integers: 1 5 fdg 32 3 4 5 6 7
	Invalid syntax. Please make sure to enter space between positive integers
	Enter 6 space-seperated integers: 1 4 5 32 41 4
	ar = [ 1 4 5 32 41 4 ]
	(1, 4)
	(1, 4)
	(4, 41)
	(41, 4)
	Number of pair(s): 4
	-- program is finished running --

Mars Messages	Run I/O
	<pre>Enter space-seperated integers array length n and divisor k: 10 3 Length (n): 10, Divisor (k): 3 Enter 10 space-seperated integers: 1      2      3 4  5 6 7 8 9 10 11 12 13 14 ar = [ 1 2 3 4 5 6 7 8 9 10 ] (1, 2) (1, 5) (1, 8) (2, 4) (2, 7) (2, 10) (3, 6) (3, 9) (4, 5) (4, 8) (5, 7) (5, 10) (6, 9) (7, 8) (8, 10) Number of pair(s): 15 -- program is finished running --</pre>
<div>Clear</div>	

Mars Messages	Run I/O
	<pre>Enter space-seperated integers array length n and divisor k: 6 3 Length (n): 6, Divisor (k): 3 Enter 6 space-seperated integers: 1 3 2 6 1 2 ar = [ 1 3 2 6 1 2 ] (1, 2) (1, 2) (3, 6) (2, 1) (1, 2) Number of pair(s): 5 -- program is finished running --</pre>
<div>Clear</div>	