

Total Marks:

Instructions:

Please read the following instructions carefully before submitting the assignment.

1. Follow instructions of TA for Submission
2. Submit your assignment in Google Classroom
3. It should be clear that your assignment will not get any credit if:
 - The assignment is submitted after the due date.
 - The assignment is copied (partial or full) from any source (websites, forums, students, etc.)

Important Note:

1. In exercises given below, subscript operator [] is not allowed to traverse the array. Use only offset notation. i.e instead of using myArray[i] use *(myArray+i) to read/write an element.
2. There shouldn't be any memory leakage in your program.
3. Make separate functions for input and output of arrays. Your main should be a sequence of function calls only
4. All allocations of 1D/2D pointers should be dynamic.
5. Delete the array when it is no longer needed.
6. All the data will be given by user.
7. Pass the pointers to functions instead of []. (It is allowed when mentioned in Question)
8. Make proper functions to solve the problems.
9. Debug your code to find errors/bugs.

Task 1:

You are given a 2D array having some elements as shown below. Your task is to remove all zero elements from the array by making a new 2D array and assign it only the non-zero elements. Assume that rows and columns of the input array are defined by the user.

2	3	1	0	0	0
0	0	0	1	1	0
1	0	0	0	0	0
1	1	1	2	0	2
5	0	0	0	10	0

Output Array: Task 2:

1	1		
1			
1	1	2	2
5	10		

2	3	1	
---	---	---	--

Write a program that creates 2D dynamic array in main function where the columns in each row should be of different length and both values should be positive. Create a function **fillArray**. This function should receive the 2D array from main function and prompt the user to provide data. Your program should only accept positive values to fill the array. Decide the remaining parameters and return type of this function at your own. Create a function **twoDimToOneDim**. This function should receive the 2D array from main function and creates a dynamic 1D array long enough to store the data of 2D array into this 1D array. This function should return the address of dynamically created 1D array to main function. Decide the remaining parameters of this function at your own. Create a function **SortArr**. This function should receive the 1D array from main function and sort its data in ascending order. Decide the remaining parameters and return type at your own. Create a function **showArr**. This function should receive the sorted 1D array and display its contents of console. Make sure that this function should not update the contents of array.

Sample Output:

Enter the size of rows: 3

Enter the columns for row#0: 4

Enter the columns for row#1: 2

Enter the columns for row#2: 3

Assume that following data is stored in 2D array:

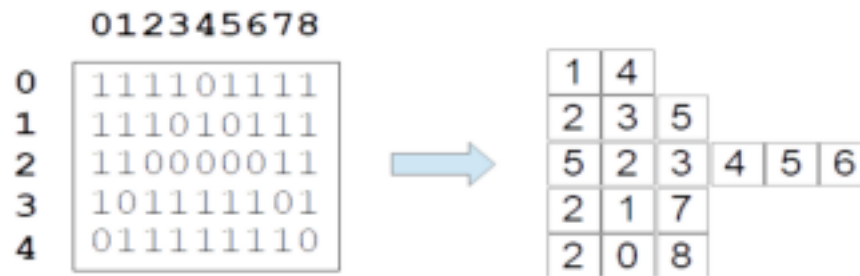
87	61	92	14
56	29		
5	78	45	

Contents of Sorted 1D array are: 5, 14, 29, 45, 56, 61, 78, 87, 92

Task 3:

Consider a black and white image that is represented as a matrix of 1's and 0's. We would like to compress the image by storing the index location of a specific value (either 0 or 1) for each row. For instance, the figure below shows transformation of an image in our desired encoding by storing location of 0's: The first cell in each row of the result shows the number of locations with

the specific value (0 or 1) in that row of the original matrix.



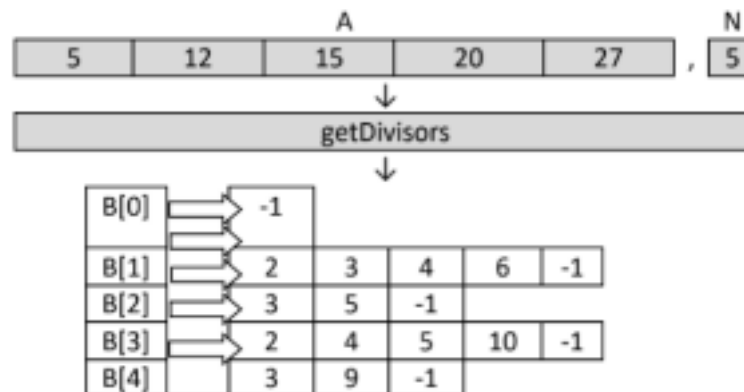
Write a function that uses the following prototype for this transformation: `int** compress(int** image, int rows, int columns, int value);` Use dynamic memory allocation for the result. The parameter value specifies the value to use for compression (either 0 or 1).

Task 4:

Write a C++ function **getDivisors** that receives an array, A, containing non-negative integers, and its size, n. The task is to compute the Divisors (other than 1 and the number itself) of all the numbers in A. The function must accomplish this task in the following way:

- All Divisors of an integer must be stored in a separate, dynamically allocated array, with -1 placed in the last index. The size of the dynamic array must exactly equal to #of Divisors+1.
- Pointers to these dynamic arrays are stored in another dynamic array (call it B) of size n. So that, when the function has finished, B[i] contains a pointer to the dynamic array containing the divisors of the number A[i], where $0 \leq i < n$.
- Lastly, the function returns B.

Following is an example input and its corresponding output, shown pictorially:



Task 5:

Write a C++ code to parse and process a given multi-dimensional character array, **arr**, which contains textual data structured as messages. The code aims to achieve the following:

Determine and count the number of messages within the array that start with the character 'A'.

Extract and store these messages in a dynamic two-dimensional character array (jagged array) while excluding the initial 'A' character and any leading spaces.

For example you are given a 2D character array (**arr**) as follows:

A	h	e	l	l	o	\0								
B	h	I	\0											
A	h	o	w		a	r	e		y	o	u	\0		
B	i		a	m		g	o	o	d	\0				
A	w	h	a	t	s		u	p	\0					
A	l	o	o	k	s		b	u	s	y	\0			

It's a chat between Person **A** and **B**. The first index of each row represents the Message sender. You need to extract the messages of Person **A** from this array and stores the characters in a new 2D dynamic character array as follows:

h	e	l	l	o	\0				
h	o	w	a	r	e	y	o	u	\0
w	h	a	t	s	u	p	\0		
l	o	o	k	s	b	u	s	y	\0

As you can see that the character **A** and **spaces** are omitted in the resultant jagged array.

Your task is to create a C++ function that will take character 2D array as input and return 2D dynamic character array. Create main as well to check the functionality.

Task 6:

Paraphrasing Application

Paraphrasing is the process of rewording a text, often done for simplification or clarity. You need to write a function that will take a character array as input and after creating a 2d string array it will return the double pointer. The prototype is given below:

string CreateDynamicArray(char ws[])** // you can pass 2d string pointer by reference here as well

Your function will receive a character array like this:

```
char ws[200]="abandon discontinue vacate#absent missing unavailable#cable wire#calculate compute
determine measure#safety security refuge";
```

In the character array there are words and their synonyms. Entries are separated by a hash symbol (#).

a) Your first task is to create a dynamic 2D strings array that will store the words and their

synonyms in different columns like the picture below.

abandon	discontinue	vacate		0	ptr0	→
absent	missing	unavailable		1	ptr1	→
cable	wire			2	ptr2	→
calculate	compute	determine	measure	3	ptr3	→
safety	security	refuge		4	ptr4	→

b) Your second task is to take input of a word and then display its last synonym

For example if user will enter absent your program will show unavailable

Note: assume that user will always enter the first word of every column to replace like abandon, absent, cable etc

c) Write proper main to check the functionality

Task 7:

ICC (International cricket council) has announced a cash prize of US (\$2000) for all those players who have achieved a milestone of **300** runs i.e., scored more than 300 runs in a recently conducted ICC T20 World cup. They have already shortlisted the teams in which there exist at-least one players who has achieved the milestone like the sample data shown in the figure 1. Now they are interested to represent the information in a more meaningful way so that they can only display the scores of those players who have achieved the required milestone. Your task is to fetch and store only the scores fulfilling the required criteria from each team in 2D dynamic array where each row should store the scores of the players from a particular country like the sample data shown in the figure 2. You have to write a function that should receive a 2D integer array named **"Scores"** (already filled with the data), rows and columns of the existing 2D array, and by reference double pointer to integer namely **"Achievers"** to store the required data. This function should allocate the memory and store the data in a 2D array (**Achievers**) from the given 2D array of **Scores**. You are supposed to allocate exactly the required amount of memory to the **Achievers** as shown in the figure 2. The data provided in the figure is just to demonstrate the problem you are required to provide generic implementation of the function. The function prototype is provided for the reference.

Scores (A 2D dynamic array of 'n' no of rows, and 'm' no of columns already filled by the user)

Figure 1:

Australia	England	India	Pakistan	New Zealand
198	352	240	286	332
362	281	290	392	218
249	408	374	346	341
426	451	160	132	412
186	195	248	482	486

Achievers (A 2D dynamic array having all the scores above 300 for each country in one row and terminating with -1 as the last element of the row)

Figure 2:

Australia	362	426	-1		
England	352	408	451	-1	
India	340	-1			
Pakistan	392	346	482	-1	
New Zealand	332	341	412	486	-1

Write appropriate functions and main to solve the above problem.