Write a program to count and find the sum of all the numbers in the array which are exactly divisible by 5 and not divisible by 2 and 3.

#include <stdio.h>

int main() {

int arr[10] = {5, 10, 15, 20, 25, 30, 35}; // declaring array

int i, j,

sum = 0,

count = 0; // initializing integers i and j for nexted loop and declating

int len = sizeof(arr) / sizeof(arr[0]); // calculating the length of the array

for (i = 0; i < len; i++) { // for loop to check each element of the array

// printf("%d",arr[i]);

if (arr[i] % 5 == 0 && arr[i] % 2 != 0 &&

arr[i] % 3 !=

0) { // to check if element is divisible by 5 and not by 2 and 3

sum += arr[i]; // if true add number to sum

count++; // if true increase count by one

}

}

printf("The sum of elements divisible by 5 and not by 2 and 3 in an array is "

"%d\n",

sum);

printf("Count of such numbers is %d", count);

}

**Write a program to check whether two given strings are an anagram.**

#include <stdio.h>

#include <string.h>

// Function to check if two strings are anagrams

int areAnagrams(char \*str1, char \*str2) {

int len1 = strlen(str1);

int len2 = strlen(str2);

// If lengths are not equal, the strings cannot be anagrams

if (len1 != len2) {

return 0; // Not anagrams

}

// Sort both strings

for (int i = 0; i < len1 - 1; i++) {

for (int j = i + 1; j < len1; j++) {

if (str1[i] > str1[j]) {

// Swap characters

char temp = str1[i];

str1[i] = str1[j];

str1[j] = temp;

}

if (str2[i] > str2[j]) {

// Swap characters

char temp = str2[i];

str2[i] = str2[j];

str2[j] = temp;

}

}

}

// Compare sorted strings

for (int i = 0; i < len1; i++) {

if (str1[i] != str2[i]) {

return 0; // Not anagrams

}

}

return 1; // Anagrams

}

int main() {

char str1[100], str2[100];

printf("Enter the first string: ");

scanf("%s", str1);

printf("Enter the second string: ");

scanf("%s", str2);

// Check if the strings are anagrams

if (areAnagrams(str1, str2)) {

printf("The given strings are anagrams.\n");

} else {

printf("The given strings are not anagrams.\n");

}

return 0;

}

**Write a program to print all unique elements in an array. For example,a[ ] =**

**{1,2,4,8,4,2,4,9,6} answer : 1,2,4,8,9,6.**

#include <stdio.h>

int main() {

int a[] = {1, 2, 4, 8, 4, 2, 4, 9, 6};

int size = sizeof(a) / sizeof(a[0]);

int isUnique;

for (int i = 0; i < size; i++) {

isUnique = 1;

for (int j = 0; j < size; j++) {

if (i != j && a[i] == a[j]) {

isUnique = 0;

break;

}

}

if (isUnique) {

printf("%d ", a[i]);

}

}

return 0;

}

**WAP reads two 2-D arrays, adds the corresponding elements and displays the**

**result on the screen.**

/\*\*WAP reads two 2-D arrays,adds the corresponding elements and displays

the result on the screen.

\*\*/

#include <stdio.h>

int main()

{

int a[2][2], b[2][2], result[2][2];//declaring three integer variables which is declaration of a 2d array

int i,j;//declaring two integer variables

// Taking input using nested for loop for matrix a

printf("Enter elements fot the first matrix\n");//asking user to enter elements for the first matrix

for ( i = 0; i < 2; ++i)//for loop for the row values for matrix a

for ( j = 0; j < 2; ++j)//for loop for the column values for matrix b

{

printf("Enter a%d%d: ", i + 1, j + 1);//displays matrix a with row column representation i.e.a to the base 11, a to the base 12 and vice versa

scanf("%d", &a[i][j]);//user input to enter any number for the 2d array of matrix a

}

// Taking input using nested for loop for matrix b

printf("\nEnter elements for the second matrix\n");//asking user to enter elements for the second matrix

for (i = 0; i < 2; ++i)//for loop for the row values of matrix b

for (j = 0; j < 2; ++j)//for loop for the column values of matrix b

{

printf("Enter b%d%d: ", i + 1, j + 1);//displays matrix b with row column representation i.e. b to the base 11, b to the base 12 and vice versa

scanf("%d", &b[i][j]);//user input to enter any number for the 2d array of matrix b

}

// adding corresponding elements of two arrays

for (i = 0; i < 2; ++i)

for (j = 0; j < 2; ++j)

{

result[i][j] = a[i][j] + b[i][j];//to calculate matrix a with corresponding value of matrix b

}

// Displaying the sum

printf("\nResultant matrix after calculation of sum:\n");//displaying a meaningful statement

for (i = 0; i < 2; ++i)

for (j = 0; j < 2; ++j)

{

printf("%d\t", result[i][j]);//displays the result after calculating sum of matrix a with the corresponding value of matrix b

if (j == 1)

printf("\n");//new line

}

return 0;

}

**\**

**4. Write a program to sort an array elements in ascending order.**

**#include <stdio.h>**

**void bubbleSort(int arr[], int n) {**

**for (int i = 0; i < n - 1; i++) {**

**for (int j = 0; j < n - i - 1; j++) {**

**// If the current element is greater than the next element, swap them**

**if (arr[j] > arr[j + 1]) {**

**int temp = arr[j];**

**arr[j] = arr[j + 1];**

**arr[j + 1] = temp;**

**}**

**}**

**}**

**}**

**int main() {**

**const int size = 6;**

**int array[size] = {64, 34, 25, 12, 22, 11};**

**printf("Original Array: ");**

**for (int i = 0; i < size; i++) {**

**printf("%d ", array[i]);**

**}**

**printf("\n");**

**// Sort the array**

**bubbleSort(array, size);**

**printf("Sorted Array (Ascending Order): ");**

**for (int i = 0; i < size; i++) {**

**printf("%d ", array[i]);**

**}**

**printf("\n");**

**return 0;**

**}**