**C++ Practical 1**

**Deadline : 30th Sept @9am**

All code must be consistently indented and variables given appropriate names.

Your name and student number must be at the top of every file.

**All code must be your own work.**

You cannot use Strings only **char** and **int** arrays**, only chars and ints.**

Example

Use cout to output to the screen. Use cin to input from the keyboard.

You must upload the code to blackboard before **30th Sept @9am**

even if not all the questions are completed. You will be unable to upload any code after that time.

1. Write a C++ function to Check if an inputted year is a Leap Year.

To determine whether a year is a leap year, follow these steps:

1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
4. The year is a leap year (it has 366 days).
5. The year is not a leap year (it has 365 days).
6. Write a C++ function to Check Whether a Number is Palindrome or Not
7. Write a C++ function to Check Whether a Number is Prime or Not. A prime number is a number only divisible by itself and 1.
8. Write a C++ function to Convert Binary Number to Decimal without using the shift operator.
9. Write a C++ function to convert chars 0-9 input from the keyboard into an integer number. Example enter in 145 as chars and convert to the int 145. The function should return 0 if any of the chars are not digits.
10. Write a nested loop that outputs the following. This nested loop contains only 2 for loops.

A

AA

AAA

AAAA

1. Write a nested loop that outputs the following. This nested loop contains only 2 for loops and at most 1 if and an else conditional statement.

A

AA

AAA

AAAA

AAA

AA

A

1. Find an element in an array and prints out the index where it was found and -1 if it was not found.

int find(int size, int arr[], int toFind)

{

return -1;

}

1. Find and return the second largest element in an array of positive integer, returns -1 if it can’t find one.

int find2ndLargest(int size, int arr[])

{

return -1;

}

1. Copy all elements from an arr1 to arr2. Both arrays are the same size.

void copyArraytoArray(int size, int arr1[], int arr2[])

{

return;

}

1. Insert an element in an array at a specified position.

Insert an element at a particular index in the array and shift

the elements further in the array to the right. Use a variable

called count to track the number of elements in the array.

The count cannot exceed the size of the array.

Return true if an element was inserted, otherwise return false.

bool insertElement(int& size, int& count, int arr[], int elementToInsert, int insertIndex)

{

return false;

}

1. Delete an element from an array of size ‘size’ at specified position.

The elements higher in the array are shifted to the left one position.

Use a variable called count to track the number of elements in the array.

The count cannot exceed the size of the array.

Return true if an element was deleted, otherwise return false.

bool deleteElement(int& size, int& count, int arr[], int deleteIndex)

{

return true;

}

1. Count the frequency of an element in an array.

int frequencyCount(int size, int arr[], int value)

{

return 0;

}

1. Count total number of duplicate elements in an array.

int countDuplicates(int size, int arr[])

{

return 0;

}

1. Find the reverse of an array. You must only declare and use 1 array

void reverse(int size, int arr[])

{

return;

}

1. Left rotate an array by one position. Note: the first element is rotated to the end of the array

int rotateLeft(int size, int arr[])

{

return -1;

}

1. You've built an inflight entertainment system with on-demand movie streaming.

Users on longer flights like to start a second movie right when their first one ends,

but they complain that the plane usually lands before they can see the ending.

So you're building a feature for choosing two movies whose total runtimes will equal the exact flight length.

The function takes an integer flightLength(in minutes) and an array of integers movieLengths(in minutes)

and returns a boolean indicating whether there are two numbers in movieLengths whose sum equals flightLength.

When building your function :

Assume your users will watch exactly two movies

Don't make your users watch the same movie twice

bool twoMovies(int flightLength, int movieLengths[], int size)

{

return false;

}

1. Takes as input an array of chars, and return the number of separate words,

where a word is one or more characters separated by spaces.

Your program should only count as words groups of characters in the ranges A..Z and a..z.

int wordCounter(int size, char characters[])

{

return 0;

}

EXTRA:

1. Write a nested loop that outputs the following. This nested loop contains only 2 for loops, and no conditional statements.

A

AA

AAA

AAAA

AAA

AA

A