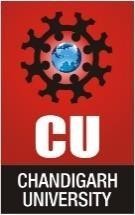
DEPARTMENT OF



COMPUTER SCIENCE & ENGINEERING

**Worksheet-2**

**Student Name:** TUSHAR GUPTA **UID:** 20BCS9268

**Branch:** CSE **Section/Group:** 607-A

**Semester:** 6th **Date of Performance:** 20/02/23

**Subject Name:** Competitive Coding-II **Subject Code:** 20CSP-351

# Aim: To implement the concept of string.

Question 1:

Given two strings needle and haystack, return the index of first occurrence of the needle in the haystack, or return -1 if needle is not present in the haystack.

# Code and output:

class Solution { public:

int strStr(string haystack, string needle) { int n = haystack.length();

int m = needle.length();

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

while(j<m && haystack[i + j] == needle[j]){ j++;

}

if(j == m) return i;

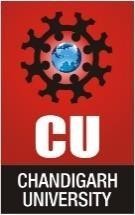
}

return -1;

}

};

DEPARTMENT OF



COMPUTER SCIENCE & ENGINEERING

# 

# Question 2 Aim:

Given two strings a and b, return the minimum number of times you should repeat string a so that string b is a substring of it. If it is impossible for b to be a substring of a then return -1.

# Code and Output:

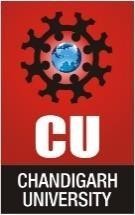
class Solution { public:

int repeatedStringMatch(string a, string b) { int n = a.length();

int m = b.length(); int k = 1;

string s = a;

DEPARTMENT OF



COMPUTER SCIENCE & ENGINEERING

while(s.length() < m){ s += a;

k++;

}

if(s.find(b) != string::npos){ return k;

}

s += a; k++;

if(s.find(b) != string::npos) { return k;

}

return -1;

}

};

