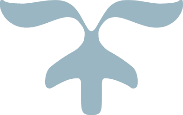


DAA WEEK – 5 SKILL – 5



# [CamelCase](https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/camelcase)

#include <stdio.h>

#include <ctype.h>

int main() {

char w[100000];

scanf("%s", w);

int len = 1;

for (int i = 0; w[i]; i++)

if (isupper(w[i]))

len++;

printf("%d", len);

return 0;

}

**Camel Case Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Strong Password](https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/strong-password)

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int minimumNumber(int x, char \*y) {

int e = 0, a = 0, b = 0, c = 0, d = 0;

const char \*f = "!@#$%^&\*()-+";

for (int g = 0; g < x; g++) {

a |= isdigit(y[g]);

b |= islower(y[g]);

c |= isupper(y[g]);

d |= !!strchr(f, y[g]);

}

e += !a + !b + !c + !d;

return e > 6 - x ? e : 6 - x;

}

int main() {

int x;

char y[101];

scanf("%d %s", &x, y);

printf("%d\n", minimumNumber(x, y));

return 0;

}

**Strong Password Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Caesar Cipher](https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/caesar-cipher-1) JAVA

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

static String caesarCipher(String s, int k) {

String t="";

for(int x=0;x<s.length();x++) {

char ch=s.charAt(x);

if((ch>=97&&ch<=122)) {

ch=(char)(ch+k);

while(ch>122) {

ch=(char)(ch-122+96);

}

} else if((ch>=65&&ch<=90)) {

ch=(char)(ch+k);

while(ch>90) {

ch=(char)(ch-90+64);

}

}

t=t+ch;

}

return t;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int n = in.nextInt();

String s = in.next();

int k = in.nextInt();

String result = caesarCipher(s, k);

System.out.println(result);

in.close();

}

}

**Caesar Cipher Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Mars Exploration](https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/mars-exploration)

#include <stdio.h>

#include <string.h>

int marsExploration(char\* s) {

int ret = 0;

int len = strlen(s);

for (int i = 0; i < len / 3; i++) {

if (s[i \* 3] != 'S') {

ret++;

}

if (s[i \* 3 + 1] != 'O') {

ret++;

}

if (s[i \* 3 + 2] != 'S') {

ret++;

}

}

return ret;

}

int main() {

char s[1001];

scanf("%s", s);

int result = marsExploration(s);

printf("%d\n", result);

return 0;

}

**Mars Exploration Test Cases**

**A screenshot of a computer

Description automatically generated**

# [HackerRank in a String!](https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/hackerrank-in-a-string) JAVA

import java.util.Scanner;

public class Solution {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int q = scan.nextInt();

while (q-- > 0) {

String str = scan.next();

System.out.println(subsequenceExists(str) ? "YES" : "NO");

}

scan.close();

}

private static boolean subsequenceExists(String str) {

String hackerrank = "hackerrank";

int index = 0;

for (int i = 0; i < str.length(); i++) {

if (str.charAt(i) == hackerrank.charAt(index)) {

index++;

}

if (index == hackerrank.length()) {

return true;

}

}

return false;

}

}

**HackerRank in a String! Test Cases**

**A screenshot of a computer

Description automatically generated**

**SKILL WEEK – 5**

[**https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/hackerrank-in-a-string**](https://www.hackerrank.com/contests/daa-skill-05-strings-and-pattern-matching-part-1/challenges/hackerrank-in-a-string)