## Q.1. Write a program to check whether a string is pangram or not

A sentence is a pangram if it contains every letter of the alphabet i.e. a-z

Test cases:

1. Input: quick brown fox jumps over the lazy dog

Output: pangram

1. Input: 123456$#Hello

Output: not a pangram

Refute Test cases:

1. Input: Jack quickly enjoyed the vibrant, mixed flavors of a cozy, warm pizza with extra cheese on top

Output: not a pangram

1. Input: Zoe quickly juggled six very large boxes full of exotic French pastries.

Output: not a pangram

Here, both the sentences in Refute Test cases have all alphabets but shows not a pangram because it exceeds 50 letters

In C, when we take input for string str[50] , it records letters till 50 characters and ignores the rest. Hence, only the 50 letters are stored and not the whole sentence since it exceeds the limit of how much this character array can store.

**Program:**

#include <stdio.h>

#include<string.h>

#include<ctype.h>

int pangram(char\*);

int pangram(char\* arr) {

int i = 0, c = arr[0];

int letter[26] = {0};

//traversal through string %[^\n]%c

for(i=0;arr[i];i++) {

c=tolower(arr[i]);

if(isalpha(arr[i]))

letter[c-'a']=1;

}

//check for 1's

for(i=0;i<26;i++) {

if(!letter[i]) return 0;

}

return 1;

}

int main()

{

char str[50];

scanf("%[^\n]s", str); //alternative to %\*[^\n]%\*c

(pangram(str)) ? printf("pangram") : printf("not a pangram");

return 0;

}

**Output:**

Input 1 :



Input 2 :



## Q.2. Write a program to convert seconds into hour:minute:second format

Take input for seconds and output the result in the format :-

Hours : Minutes : Seconds

Test cases:

1. Input: 7845

Output: 2 Hours 10 Minutes 45 Seconds

1. Input: 9871

Output: 2 Hours 45 Minutes 31 Seconds

Refute Test cases:

1. Input: 7836.24

Output: 2 Hours : 10 Minutes : 36 Seconds

1. Input: Hello&

Output: 1166 Hours : 56 Minutes : 48 Seconds

In C, since seconds takes input in the form of integer data type.

If the inputs are any data types other than integers then, the seconds remain uninitialized and get assigned a garbage value.

Hence why, we get output for the above refute cases without errors as seconds is assigned to garbage value and the function calculates according to the assigned garbage value

**Program:**

#include<stdio.h>

int main() {

int seconds;

int hr,min,sec;

scanf("%d",&seconds);

hr=seconds/3600;

min=((seconds%3600)/60);

sec=((seconds%3600)%60);

printf("%d Hours : %d Minutes : %d Seconds",hr,min,sec);

return 0;

}

**Output:**

Input 1 :



Input 2 :

