

Experiment 7

String

Student Name: Bhanu Pundir

UID: 20BCS1439

Branch: BE CSE

Section/Group: 620-B

Semester: 5th

Date of performance: 28 Oct 2022

Subject Name: CC Lab

Subject Code: 20CSP-314

1. Aim/Overview of the practical:

To Demonstrate the concept of String Data Structure

A numeric string, , is beautiful if it can be split into a sequence of two or more positive integers, , satisfying the following conditions:

<https://www.hackerrank.com/challenges/separate-the-numbers/problem?isFullScreen=true>

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

3. Objective:

- To understand the concept of String Data Structure.
- To implement the concept of String Data Structure.
- To learn different approaches used to separate the numbers.

4. Code:

```
#include <bits/stdc++.h>

#define ll long long
#define ld double
#define pii pair <ll, ll>
```



```
#define mp make_pair

using namespace std;

int main() {
    int q;

    cin >> q;

    while (q--) {
        string s;

        cin >> s;

        if (s[0] == '0') {
            cout << "NO\n";
            continue;
        }

        ll now = 0;

        bool st = false;

        for (int i = 0; i < (int)s.size(); i++) {
            now *= 10;
            now += s[i] - '0';

            ll res = 0;

            if (s[i + 1] == '0') {
                continue;
            }

            int cnt = 1;
```

```
for (int j = i + 1; j < (int)s.size(); j++) {
    res *= 10;
    res += s[j] - '0';

    if (res == now + cnt) {
        if (j + 1 == (int)s.size()) {
            st = true;
            break;
        }

        if (s[j + 1] == '0') {
            break;
        }

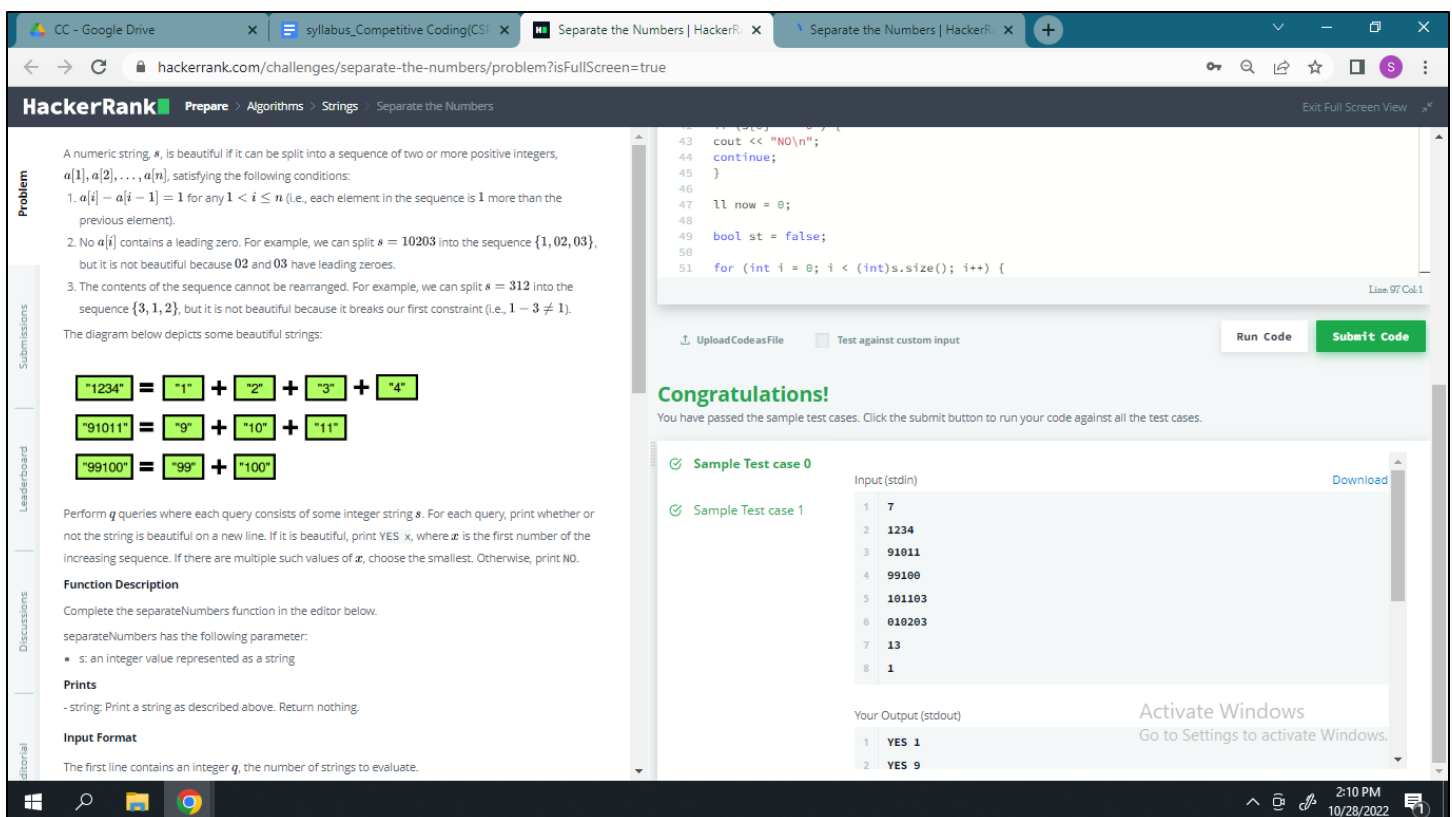
        res = 0;
        cnt++;
    }

    if (st) {
        break;
    }

    if (st) {
        cout << "YES " << now << endl;
    } else {
        cout << "NO\n";
    }
}

return 0;
}
```

5. Result/Output/Writing Summary:



The screenshot shows the HackerRank interface for the 'Separate the Numbers' problem. The problem description states that a numeric string s is beautiful if it can be split into a sequence of two or more positive integers $a[1], a[2], \dots, a[n]$, satisfying the following conditions:

- $a[i] - a[i - 1] = 1$ for any $1 < i \leq n$ (i.e., each element in the sequence is 1 more than the previous element).
- No $a[i]$ contains a leading zero. For example, we can split $s = 10203$ into the sequence $\{1, 02, 03\}$, but it is not beautiful because 02 and 03 have leading zeroes.
- The contents of the sequence cannot be rearranged. For example, we can split $s = 312$ into the sequence $\{3, 1, 2\}$, but it is not beautiful because it breaks our first constraint (i.e., $1 - 3 \neq 1$).

The diagram below depicts some beautiful strings:

```

"1234" = "1" + "2" + "3" + "4"
"91011" = "9" + "10" + "11"
"99100" = "99" + "100"
  
```

Perform q queries where each query consists of some integer string s . For each query, print whether or not the string is beautiful on a new line. If it is beautiful, print YES x , where x is the first number of the increasing sequence. If there are multiple such values of x , choose the smallest. Otherwise, print NO.

Function Description

Complete the `separateNumbers` function in the editor below.

`separateNumbers` has the following parameter:

- s : an integer value represented as a string

Prints

- string: Print a string as described above. Return nothing.

Input Format

The first line contains an integer q , the number of strings to evaluate.

The code editor shows a C++ solution:

```

43 cout << "NO\n";
44 continue;
45 }
46
47 ll now = 0;
48
49 bool st = false;
50
51 for (int i = 0; i < (int)s.size(); i++) {
  
```

The test results show that the solution passed all sample test cases:

Sample Test case	Input (stdin)	Your Output (stdout)
Sample Test case 0	7	YES 1
Sample Test case 1	1234 91011 99100 101103 010203 13 1	YES 9



CC - Google Drive x syllabus_Competitive Coding(CS) x Separate the Numbers | HackerR x Separate the Numbers | HackerR x +

hackerrank.com/challenges/separate-the-numbers/problem?isFullScreen=true

HackerRank Prepare Algorithms > Strings Separate the Numbers Exit Full Screen View

Problem

1234
91011
99100
101103
010203
13
1

Sample Output 0

YES 1
YES 9
YES 99
NO
NO
NO
NO

Submissions

Leaderboard

Discussions

Editorial

Explanation 0

The first three numbers are beautiful (see the diagram above). The remaining numbers are not beautiful:

- For $s = 101103$, all possible splits violate the first and/or second conditions.
- For $s = 010203$, it starts with a zero so all possible splits violate the second condition.
- For $s = 13$, the only possible split is $\{1, 3\}$, which violates the first condition.
- For $s = 1$, there are no possible splits because s only has one digit.

Sample Input 1

4
99910001001
7891011
9899100
999100010001

Sample Output 1

```
43 cout << "NO\n";  
44 continue;  
45 }  
46  
47 ll now = 0;  
48  
49 bool st = false;  
50  
51 for (int i = 0; i < (int)s.size(); i++) {
```

Line 97 Col:1

Upload Code as File Test against custom input Run Code Submit Code

Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0
Sample Test case 1

Expected Output

1 YES 1
2 YES 9
3 YES 99
4 NO
5 NO
6 NO
7 NO

Activate Windows
Go to Settings to activate Windows.

2:11 PM
10/28/2022



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

**NAAC
GRADE A+**
ACCREDITED UNIVERSITY

CC - Google Drive x syllabus_Competitive Coding(CS) x Separate the Numbers | HackerR x Separate the Numbers | HackerR x

hackerrank.com/challenges/separate-the-numbers/problem?isFullScreen=true

HackerRank Prepare Algorithms Strings Separate the Numbers Exit Full Screen View

Problem

Sample Output 0

```
YES 1
YES 9
YES 99
NO
NO
NO
NO
NO
```

Explanation 0

The first three numbers are beautiful (see the diagram above). The remaining numbers are not beautiful:

- For $s = 101103$, all possible splits violate the first and/or second conditions.
- For $s = 010203$, it starts with a zero so all possible splits violate the second condition.
- For $s = 13$, the only possible split is $\{1, 3\}$, which violates the first condition.
- For $s = 1$, there are no possible splits because s only has one digit.

Sample Input 1

```
4
99910001001
7891011
9899100
999100010001
```

Sample Output 1

```
YES 999
YES 7
YES 98
NO
```

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!
You are now 215 points away from the 4th star for your problem solving badge. 22% 260/475

Congratulations
You solved this challenge. Would you like to challenge your friends? f t in

Next Challenge

Test case 0 Test case 1 Test case 2 Test case 3 Test case 4 Test case 5 Test case 6

Compiler Message
Success

Input (stdin)
1 7
2 1234
3 91011
4 99100
5 101103
6 010203
7 13
8 1

Activate Windows
Go to Settings to activate Windows.

2:11 PM
10/28/2022

The screenshot shows the HackerRank interface for the 'Separate the Numbers' challenge. The problem description states that a numeric string s is beautiful if it can be split into a sequence of two or more positive integers $a[1], a[2], \dots, a[n]$ satisfying the following conditions:

- $a[i] - a[i - 1] = 1$ for any $1 < i \leq n$ (i.e., each element in the sequence is 1 more than the previous element).
- No $a[i]$ contains a leading zero. For example, we can split $s = 10203$ into the sequence $\{1, 02, 03\}$, but it is not beautiful because 02 and 03 have leading zeroes.
- The contents of the sequence cannot be rearranged. For example, we can split $s = 312$ into the sequence $\{3, 1, 2\}$, but it is not beautiful because it breaks our first constraint (i.e., $1 - 3 \neq 1$).

The diagram below depicts some beautiful strings:

```

"1234" = "1" + "2" + "3" + "4"
"91011" = "9" + "10" + "11"
"99100" = "99" + "100"
  
```

Perform q queries where each query consists of some integer string s . For each query, print whether or not the string is beautiful on a new line. If it is beautiful, print YES x , where x is the first number of the increasing sequence. If there are multiple such values of x , choose the smallest. Otherwise, print NO.

Function Description

Complete the `separateNumbers` function in the editor below.

`separateNumbers` has the following parameter:

- s : an integer value represented as a string

Prints

- string: Print a string as described above. Return nothing.

Input Format

The first line contains an integer q , the number of strings to evaluate.

Test Cases:

Test Case	Expected Output
Test case 0	YES 1
Test case 1	YES 9
Test case 2	YES 99
Test case 3	NO
Test case 4	NO
Test case 5	NO
Test case 6	NO

Experiment 7.2

1. Aim/Overview of the practical:

To Demonstrate the concept of String Data Structure

A pangram is a string that contains every letter of the alphabet. Given a sentence determine whether it is a pangram in the English alphabet. Ignore case. Return either pangram or not pangram as appropriate.

<https://www.hackerrank.com/challenges/pangrams/problem?isFullScreen=true>

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

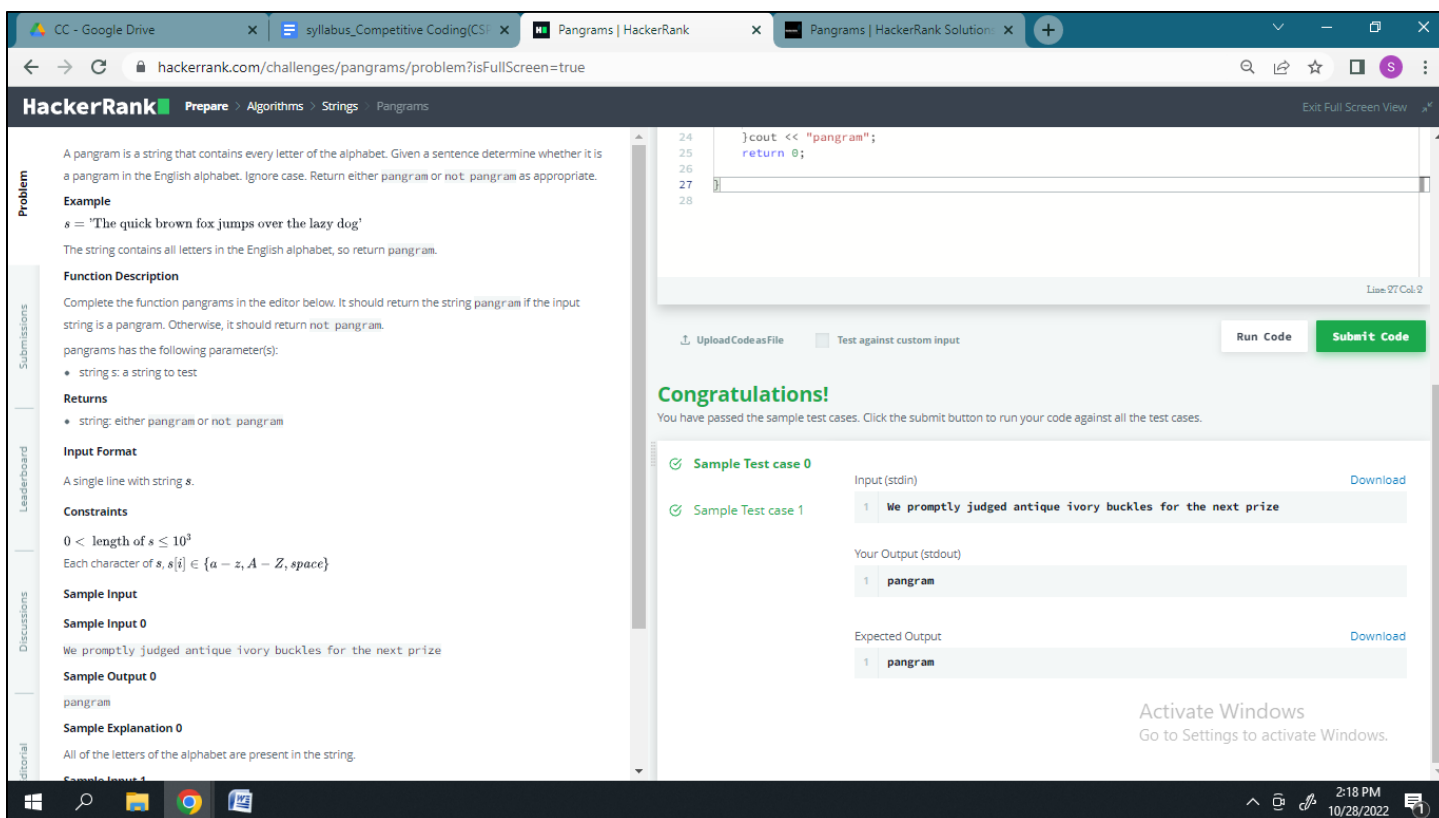
3. Objective:

- To understand the concept of String Data Structure.
- To implement the concept of String Data Structure.
- To learn different approaches used to separate the numbers.

4. Code:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    string a; getline(cin, a); map <char,int> he;
    for (int g=0;g<a.length(); g++)
    {
        if (a[g]>='A' && a[g]<='Z')
        {
            a[g]=char(a[g]- 'A'+ 'a');
            he[a[g]]++;
        }
        if (a[g]>='a' && a[g]<='z')
        {
            he[a[g]]++;
        }
    }
    for (int g=0; g<26; g++)
    {
        if (!he[char('a'+g)])
        {
            cout << "not pangram"; return 0;
        }
    }
    cout << "pangram";
    return 0;
}
```


5. Result/Output/Writing Summary:



The screenshot shows a web browser window with the HackerRank website. The browser tabs include 'CC - Google Drive', 'syllabus_Competitive Coding(CS)', 'Pangrams | HackerRank', and 'Pangrams | HackerRank Solution:'. The address bar shows 'hackerrank.com/challenges/pangrams/problem?isFullScreen=true'. The page title is 'HackerRank' and the breadcrumb is 'Prepare > Algorithms > Strings > Pangrams'. The left sidebar contains links for 'Problem', 'Submissions', 'Leaderboard', 'Discussions', and 'Editorial'. The main content area displays the problem description for 'Pangrams', which asks to determine if a given string is a pangram (contains every letter of the alphabet). It includes an example, function description, constraints, and sample input/output. The code editor shows a C++ solution:

```
24 }cout << "pangram";  
25 return 0;  
26  
27  
28
```

. Below the code editor, there are buttons for 'Upload Code as File', 'Test against custom input', 'Run Code', and 'Submit Code'. A green banner says 'Congratulations! You have passed the sample test cases. Click the submit button to run your code against all the test cases.' Below this, there are two sample test cases, both marked as passed. The first sample test case shows the input 'We promptly judged antique ivory buckles for the next prize' and the output 'pangram'. The second sample test case shows the input 'pangram' and the output 'pangram'. At the bottom right, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.'



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



CC - Google Drive x syllabus_Competitive Coding(CS) x Pangrams | HackerRank x Pangrams | HackerRank Solution: x +

hackerrank.com/challenges/pangrams/problem?isFullScreen=true

HackerRank Prepare > Algorithms > Strings > Pangrams Exit Full Screen View

Problem

A pangram is a string that contains every letter of the alphabet. Given a sentence determine whether it is a pangram in the English alphabet. Ignore case. Return either `pangram` or `not pangram` as appropriate.

Example

`s = 'The quick brown fox jumps over the lazy dog'`

The string contains all letters in the English alphabet, so return `pangram`.

Function Description

Complete the function `pangrams` in the editor below. It should return the string `pangram` if the input string is a pangram. Otherwise, it should return `not pangram`.

`pangrams` has the following parameter(s):

- string `s`: a string to test

Returns

- string: either `pangram` or `not pangram`

Input Format

A single line with string `s`.

Constraints

- $0 < \text{length of } s \leq 10^3$
- Each character of `s`, `s[i] ∈ {a - z, A - Z, space}`

Sample Input 0

We promptly judged antique ivory buckles for the next prize

Sample Output 0

pangram

Sample Explanation 0

All of the letters of the alphabet are present in the string.

Submissions

Leaderboard

Discussions

Editorial

Upload Code as File Test against custom input Run Code Submit Code

You have earned 20.00 points!
You are now 195 points away from the 4th star for your problem solving badge.

29% 280/475

Congratulations
You solved this challenge. Would you like to challenge your friends? f t in

Next Challenge

Test case 0 Test case 1 Test case 2 Test case 3 Test case 4 Test case 5 Test case 6

Compiler Message
Success

Input (stdin)
1 We promptly judged antique ivory buckles for the next prize

Expected Output
1 pangram

Activate Windows
Go to Settings to activate Windows.

2:18 PM 10/28/2022

HackerRank Prepare > Algorithms > Strings > Pangrams

Complete the function pangram which takes a string s as input and returns a boolean value. A string is a pangram if it contains every letter of the alphabet at least once. Otherwise, it should return not pangram.

pangrams has the following parameter(s):

- string s: a string to test

Returns

- string: either pangram or not pangram

Input Format

A single line with string s.

Constraints

- $0 < \text{length of } s \leq 10^3$
- Each character of s, $s[i] \in \{a - z, A - Z, \text{space}\}$

Sample Input

Sample Input 0

We promptly judged antique ivory buckles for the next prize

Sample Output 0

pangram

Sample Explanation 0

All of the letters of the alphabet are present in the string.

Sample Input 1

We promptly judged antique ivory buckles for the prize

Sample Output 1

not pangram

Sample Explanation 1

The string lacks an x.

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main()
4 {
5     string a; getline(cin, a); map<char, int> he;
6     for (int g=0; g<a.length(); g++)
7     {
8         if (a[g]>='A' && a[g]<='Z')
9         {
10             a[g]=char(a[g]-'A'+ 'a');
11             he[a[g]]++;
12         }
13         if (a[g]>='a' && a[g]<='z')
14         {
15             he[a[g]]++;
16         }
17     }
18     for (int g=0; g<26; g++)
19     {
20         if (!he[char('a'+g)])
21         {
22             cout << "not pangram"; return 0;
23         }
24     }cout << "pangram";
25     return 0;
26 }
27
28

```

Activate Windows
Go to Settings to activate Windows.

Run Code Submit Code

2:19 PM
10/28/2022

Learning outcomes (What I have learnt):

- Learned about the concept of String Data Structure.
- Learned about implementing the concept of String Data Structure.
- Learned different approaches used to separate the numbers.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			