Bangladesh University of Business and Technology
Department of Computer Science and Engineering

# CSE 232: Data Structures Lab

Solution to Lab 03 Tasks

## Task 1

Write a program that will imitate the property of DELETING. The program will take a *text*, *position* and *length* as input, delete the characters at the given *position*, and print the *text* after deletion.

| **Sample Input:**<br>IRONMAN<br>3<br>3 | **Sample Output:**<br>IRON |
|---|---|
| **Sample Input:**<br>SPIDERMAN<br>-1<br>3 | **Sample Output:**<br>SPIDERMAN |

Sample Solution:

```cpp
#include <iostream>
using namespace std;

string SUBSTRING(string str, int initial, int length)
{
    string substr = str.substr(initial, length);
    return substr;
}

string DELETE(string text, int position, int length)
{
    string substr1, substr2;
    substr1 = SUBSTRING(text, 0, position);
    substr2 = SUBSTRING(text, position + length, text.size() - position - length);
    return substr1 + substr2;
}

int main()
{
    string str;
    int p, l;
    cin >> str >> p >> l;
    if (p < 0 || p > str.size())
        cout << str << endl;
    else
        cout << DELETE(str, p, l) << endl;
}
```

## Task 2

Write a program that will imitate the property of INDEXING. The program will take a *text* and *pattern* as input, and print the position where the pattern first appears in the text.
Note: If the pattern does not appear in the text, print "-1".
(For understanding purposes, in the third example, an underscore (_ ) is used to indicate a space. For your program, you should just input the pattern with spaces, not use an underscore ( _ ).

| | |
|---|---|
| **Sample Input:**<br>HIS FATHER IS THE PROFESSOR<br>THE | **Sample Output:**<br>6 |
| **Sample Input:**<br>HIS FATHER IS THE PROFESSOR<br>THEN | **Sample Output:**<br>-1 |
| **Sample Input:**<br>HIS FATHER IS THE PROFESSOR<br>_THE_ | **Sample Output:**<br>13 |

Sample Solution:

```
#include <iostream>
using namespace std;

int INDEX(string text, string pattern)
{
    int i;
    for (i = 0; i < (int) text.size(); i++)
    {
        bool flag = true;
        for (int j = 0; j < (int) pattern.size(); j++)
        {
            if (text[i + j] == pattern[j])
                continue;
            else
            {
                flag = false;
                break;
            }
        }
        if (flag == false)
            continue;
        else
            return i;
    }
    if (i == (int) text.size())
        return -1;
}

int main()
{
    string t, p;
    getline(cin, t);
    getline(cin, p);
    cout << INDEX(t, p) << endl;
}
```

## Task 3

A word or sentence is called a *Pangram* if all the characters of the alphabet appear in it at least once. Write a program that will take a string as input and check whether it is a *Pangram* or not. The input string may contain both uppercase and lowercase letters.

| **Sample Input:**<br>asmallword | **Sample Output:**<br>NO |
|---|---|
| **Sample Input:**<br>TheQuickBrownFoxJumpsOverTheLazyDog | **Sample Output:**<br>YES |

Sample Solution:

```
#include <iostream>
using namespace std;

int main()
{
    string str;
    int cnt = 0;
    cin >> str;
    for (int i = 0; i < (int) str.size(); i++)
    {
        if (str[i] >= 'A' && str[i] <= 'Z')
            str[i] = str[i] + 32;
    }

    for (int i = 'a'; i <= 'z'; i++)
    {
        for (int j = 0; j < (int) str.size(); j++)
        {
            if (str[j] == i)
            {
                cnt++;
                break;
            }
        }
    }

    if (cnt == 26)
        cout<<"YES"<<endl;
    else
        cout<<"NO"<<endl;
}
```