

# IT300 – Design and Analysis of Algorithms

## Lab Assignment – 6

**Name:** Niraj Nandish

**Roll No:** 191IT234

1. Program 1
  - a. Code

```
#include <iostream>
#include <queue>
#include <set>
#include <stdio.h>

using namespace std;

int shortestChainLength(string start, string end, set<string> wordDict)
{
    if (start == end || wordDict.find(end) == wordDict.end())
    {
        return 0;
    }

    int lvl = 0, wLen = start.size(), size;
    string word;
    char orig_char;
    queue<string> Q;

    Q.push(start);
    while (!Q.empty())
    {
        lvl++;
        size = Q.size();
        for (int i = 0; i < size; i++)
        {
            word = Q.front();
            Q.pop();
            for (int pos = 0; pos < wLen; ++pos)
            {
                orig_char = word[pos];
                for (char c = 'a'; c <= 'z'; ++c)
                {
                    word[pos] = c;
                    if (word == end)
                    {
                        return lvl + 1;
                    }
                }
            }
            if (wordDict.find(word) == wordDict.end())
                Q.push(word);
        }
    }
}
```

```

        {
            continue;
        }
        Q.push(word);
        wordDict.erase(word);
    }
    word[pos] = orig_char;
}
}
}

return 0;
}

int main()
{
    int n;
    string start, end, temp;
    set<string> wordDict;

    cout << "Enter number of words: ";
    cin >> n;
    cout << endl;

    for (int i = 0; i < n; i++)
    {
        cout << "Enter word " << i + 1 << ": ";
        cin >> temp;
        wordDict.insert(temp);
    }

    cout << "\nEnter starting word: ";
    cin >> start;
    cout << "Enter ending word: ";
    cin >> end;
    cout << "\nLength of shortest chain of words: " << shortestChainLength(start, end, wordDict) <<
endl;

    return 0;
}

```

b. Screenshots

```
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → g++ prog1.cpp
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → ./a.out
Enter number of words: 7

Enter word 1: hit
Enter word 2: cog
Enter word 3: hot
Enter word 4: dot
Enter word 5: dog
Enter word 6: lot
Enter word 7: log

Enter starting word: hit
Enter ending word: dog

Length of shortest chain of words: 4
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 →
```

## 2. Program 2

### a. Code

```
#include <iostream>
#include <stdio.h>
#include <vector>

using namespace std;

bool compFunc(string x, string y)
{
    string YX = y.append(x);
    string XY = x.append(y);
    return XY.compare(YX) > 0 ? 1 : 0;
}

void maxSalary(vector<string> inpArr)
{
    sort(inpArr.begin(), inpArr.end(), compFunc);
    for (int i = 0; i < inpArr.size(); i++)
    {
        cout << inpArr[i];
    }
}

int main()
{
    int n, inp[n];
    string temp;
    vector<string> inpArr;

    cout << "Enter number of numbers: ";
    cin >> n;

    for (int i = 0; i < n; i++)
    {
        cout << "Enter number " << i + 1 << ": ";
        cin >> temp;
        inpArr.push_back(temp);
    }

    cout << "\nMaximum salary: ";
    maxSalary(inpArr);
    cout << endl;

    return 0;
}
```

b. Screenshots

```
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → g++ prog2.cpp
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → ./a.out
Enter number of numbers: 2
Enter number 1: 2
Enter number 2: 21

Maximum salary: 221
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → ./a.out
Enter number of numbers: 4
Enter number 1: 23
Enter number 2: 39
Enter number 3: 92
Enter number 4: 991

Maximum salary: 991923923
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 →
```

### 3. Program 3

#### a. Code

```
#include <iostream>
#include <stdio.h>

using namespace std;

void minRefills(int d, int m, int n, int gas[])
{
    int refills = 0, curr = 0, limit = m;
    while (limit < d)
    {
        if (curr >= n || gas[curr] > limit)
        {
            cout << "-1" << endl;
            return;
        }
        while (curr < n - 1 && gas[curr + 1] <= limit)
        {
            curr++;
        }
        refills++;
        limit = gas[curr] + m;
        curr++;
    }
    cout << refills << endl;
}

int main()
{
    int d, m, n, gas[n];
    cout << "Enter distance of your destination: ";
    cin >> d;
    cout << "Enter max distance your car can travel on a full tank: ";
    cin >> m;
    cout << "Enter number of gas stations: ";
    cin >> n;
    cout << endl;

    for (int i = 0; i < n; i++)
    {
        cout << "Enter distance of gas station " << i + 1 << ": ";
        cin >> gas[i];
    }

    cout << "\nMinimum number of refills: ";
    minRefills(d, m, n, gas);

    return 0;
}
```

b. Screenshots

```
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → g++ prog3.cpp
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → ./a.out
Enter distance of your destination: 950
Enter max distance your car can travel on a full tank: 400
Enter number of gas stations: 4

Enter distance of gas station 1: 200
Enter distance of gas station 2: 375
Enter distance of gas station 3: 550
Enter distance of gas station 4: 750

Minimum number of refills: 2
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 → ./a.out
Enter distance of your destination: 10
Enter max distance your car can travel on a full tank: 3
Enter number of gas stations: 3

Enter distance of gas station 1: 1
Enter distance of gas station 2: 5
Enter distance of gas station 3: 9

Minimum number of refills: -1
niraj ~/Desktop/IT-Labs/DAA-Lab/Lab7 →
```