**IT300 – Design and Analysis of Algorithms**

Lab Assignment – 6

**Name:** Niraj Nandish

**Roll No:** 191IT234

1. Program 1
   1. 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())

{

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;

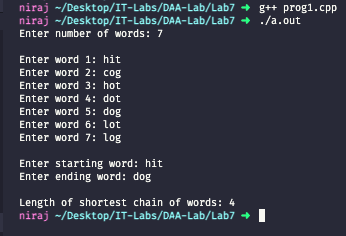
cout << "Enter ending word: ";

cin >> end;

cout << "\nLength of shortest chain of words: " << shortestChainLength(start, end, wordDict) << endl;

return 0;

}

* 1. Screenshots

1. Program 2
   1. 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);

}

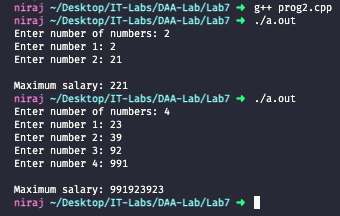
cout << "\nMaximum salary: ";

maxSalary(inpArr);

cout << endl;

return 0;

}

* 1. Screenshots

1. Program 3
   1. 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;

}

* 1. Screenshots

A screenshot of a computer

Description automatically generated with medium confidence