Experiment 4: You are in the process of creating a text editor (like notepad /Ms Word). The text editor saves particular word in form a string. We need to provide the feature of Find and Replace a particular substring in the given text editor.

CO Attained: CO2 and CO4

#include <stdio.h>

#include <math.h>

#include <stdbool.h>

int calc\_hash(char \*arr, int a, int n){

int pm = 2;

int ans = 0;

int k = n;

for(int i=a ; i<n+a ; i++){

ans += (arr[i])\*(pow(pm,(k--)-1));

}

return ans;

}

int main()

{

char str[] = "CABDCAC";

char pat[] = "ABD";

int n = sizeof(str)/sizeof(str[0])-1;

int m = sizeof(pat)/sizeof(pat[0])-1;

int hash = calc\_hash(pat,0,m);

printf("Pattern Hash : %d\n",hash);

bool flag = false;

for(int i=0 ; i<=n-m ; i++){

int curr\_hash = calc\_hash(str,i,m);

if(hash == curr\_hash){

flag = true;

break;

}

}

if(flag)printf("Found");

else printf("Not Found");

return 0;

}

Experiment 5: Characters are sorted in a form of a list (non continuous). We have to check if the list of characters has a given character in it or not.

CO Attained: CO3, CO4 and CO5

#include<bits/stdc++.h>

using namespace std;

class Node{

public:

int val;

Node \*next;

Node(){

this->val = INT\_MIN;

this->next = NULL;

}

Node(int x){

val = x;

this->next = NULL;

}

};

class LL{

Node \*head;

public:

LL(){

this->head = NULL;

}

void insertAtBegin(int x){

Node \*n1 = new Node(x);

n1->next = head;

head = n1;

}

void insertAtEnd(int x){

Node \*n1 = new Node(x);

if(head==NULL){

head=n1;

return ;

}

Node \*temp = head;

while(temp->next){

temp = temp->next;

}

temp->next = n1;

}

bool isFound(int x){

Node \*temp = head;

while(temp){

if(temp->val == x){

return true;

}

temp = temp->next;

}

return false;

}

void printList(){

if(!head){

cout<<"Underflow: No data to print.";

return ;

}x`

Node \*temp = head;

while(temp){

cout<<temp->val<<"->";

temp = temp->next;

}

}

};

int main(){

LL l1,l2,l3;

l1.insertAtEnd(1);

l1.insertAtEnd(5);

l1.insertAtEnd(7);

cout<<(l1.isFound(12) ? "Found\n" : "Not Found\n");

return 0;

}