National

University

of

Computer

and

Emerging

Sciences

Chiniot-Faisalabad

Campus



**CL2001 – Data Structures - Lab**

**Lab 8**

**“Recursion & Basics of Tree’s”**

**Fall 2024**

**Maximum Marks:** 100 **Due Date:** 11 October 2024

**Submitted By**

|  |  |
| --- | --- |
| Name: | Talha Abdullah |
| Student ID: | 22F-3194 |
| Section: | BAI-3A |

**Submitted To**

Muhammad Yousaf

Lecturer – AI & Data Science Department

**Submission Date**

11 October 2024

# **Problem 1**

## **Source Code**

#include<iostream>

#include<string>

using namespace std;

void copyStr(string s1, string& s2, int index);

int magicNumber(int arr[], int size);

int main() {

string s1, s2;

int size = 0, output = 0, val = 0, index = 0;

int arr[100];

while (true) {

cout << "-----------Menu--------------" << endl;

cout << "1. to Copy the string" << endl;

cout << "2. to magic the numbers" << endl;

cout << "0. to exit the program" << endl;

cout << endl;

cout << "Enter a value to visit the menu: ";

cin >> val;

cin.ignore();

switch (val)

{

case 1:

cout << "Enter a string: ";

getline(cin, s1);

s2 = "";

index = s1.length();

copyStr(s1, s2, index - 1);

cout << "\nThe output of the string is: " << s2 << endl;

break;

case 2:

cout << "Enter the size of an array: ";

cin >> size;

cout << "Enter " << size << " elements: ";

for (int i = 0; i < size; i++) {

cin >> arr[i];

}

output = magicNumber(arr, size);

cout << "\nThe output is: " << output << endl;

break;

case 0:

return 0;

default:

cout << "Invalid choice! Try again.\n";

break;

}

}

return 0;

}

void copyStr(string s1, string& s2, int index) {

if (index < 0) {

return;

}

else {

if (isupper(s1[index])) {

s2 += s1[index];

}

copyStr(s1, s2, index - 1);

}

}

int magicNumber(int arr[], int size) {

if (size < 0) {

return 0;

}

else {

if (size % 2 == 0) {

return arr[size] \* 9 + magicNumber(arr, size - 1);

}

else {

return arr[size] \* 99 + magicNumber(arr, size - 1);

}

}

}

## **Screenshot**

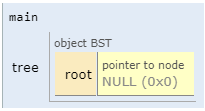
A screenshot of a computer program

Description automatically generated

# **Problem 4**

## **Screenshot**

**(A)**



A screenshot of a computer

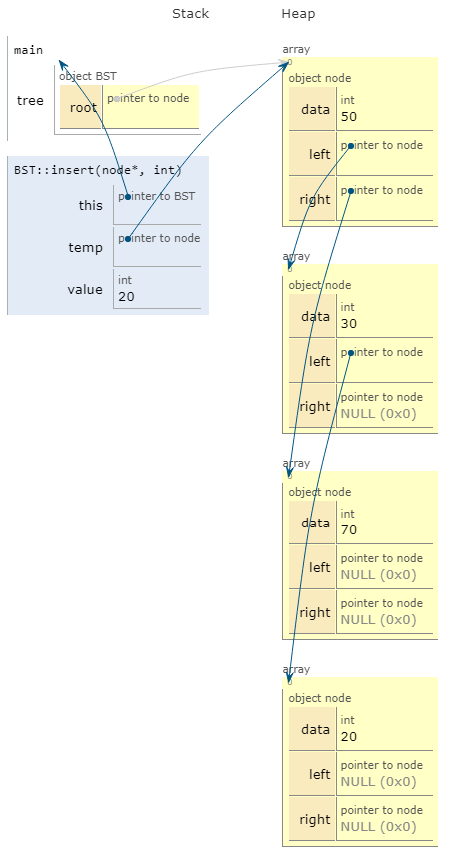
Description automatically generated

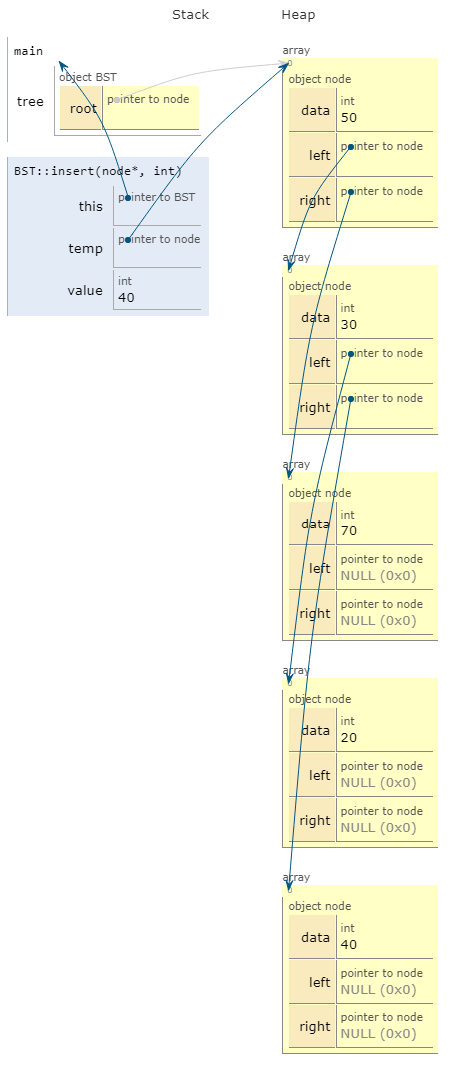
A screenshot of a computer

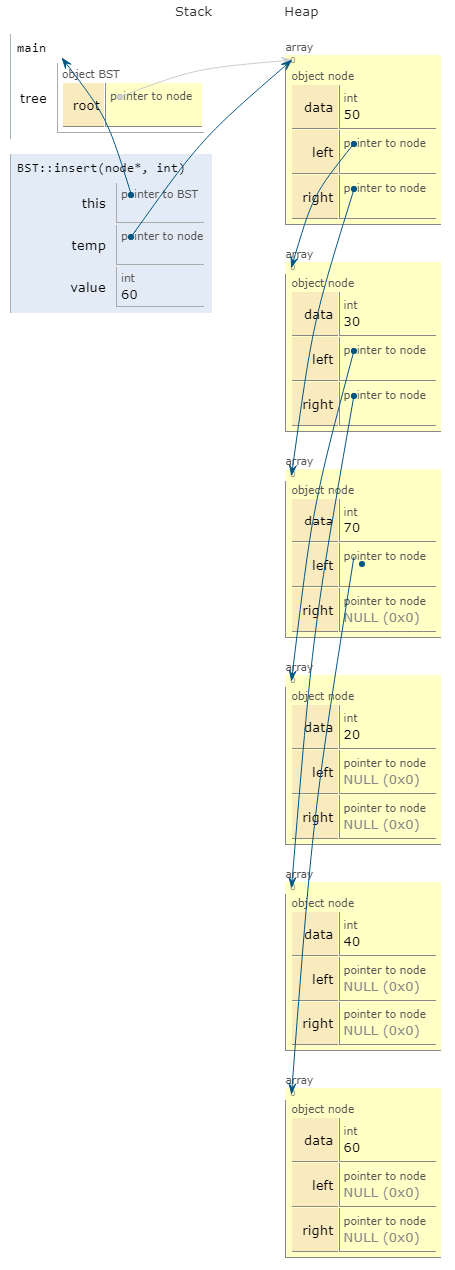
Description automatically generated

A screenshot of a computer program

Description automatically generated







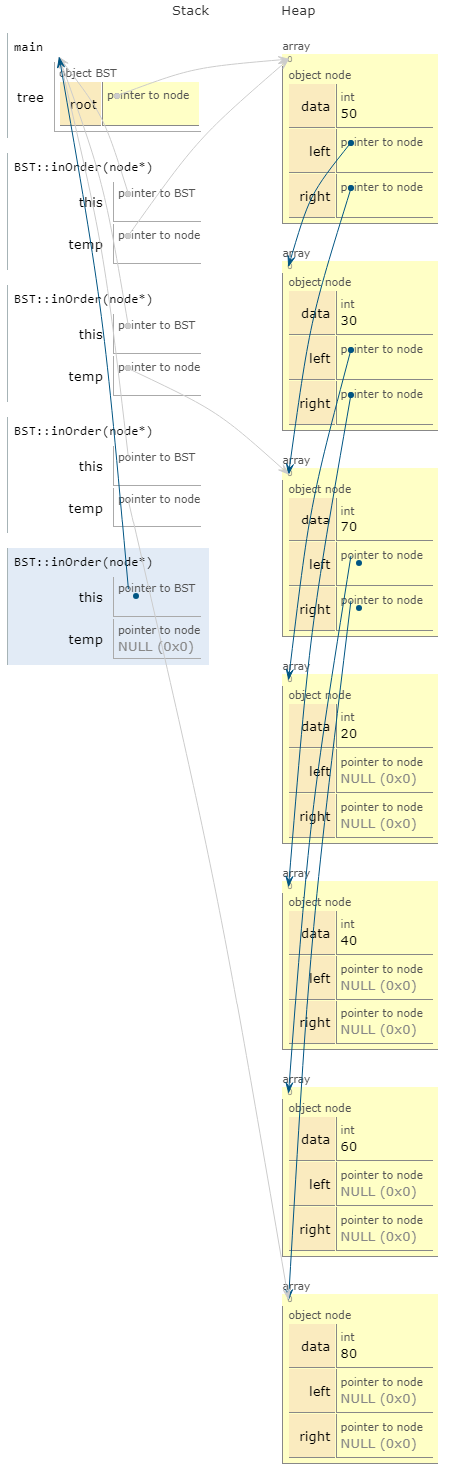
A screenshot of a computer

Description automatically generated

**(B)**

A screenshot of a computer

Description automatically generated



# **Problem 3**

## **Formula solution**

**(1)**

Node=127

H=log2(127+1)-1

H=log2(128)-1

H=7-1

H=6

**(2)**

H=5

Leafnode=12

**Inter node**

I=l-1

L=12-1

L=11

**Number of node**

N=I+L

N=11+12

N=23

**(3)**

H=6

Inter node= 2^n-1 = 2^6-1 = 63

Leaf Node = 2^h = 2^6 = 64

Total node = 2^n+1 -1 = 2^7 -1 = 127

**(4)**

N=512

H=log2(n+!-1)

H=log2(512)-1

H=9-1

H=8

**(5)**

# **Problem 2**

## **Source Code**

#include <iostream>

using namespace std;

class node

{

public:

int data;

node\* left;

node\* right;

node(int val)

{

data = val;

left = right = nullptr;

}

};

class BST

{

public:

node\* root;

BST() : root(nullptr) {}

void insert(node\*& temp, int value)

{

if (temp == nullptr) {

temp = new node(value);

return;

}

if (value < temp->data)

{

insert(temp->left, value);

}

else if (value > temp->data)

{

insert(temp->right, value);

}

}

bool find(node\* temp, int key)

{

while (temp != nullptr)

{

if (temp->data == key)

{

return true;

}

if (key < temp->data)

{

temp = temp->left;

}

else

{

temp = temp->right;

}

}

return false;

}

};

int main()

{

BST tree;

tree.insert(tree.root, 50);

tree.insert(tree.root, 30);

tree.insert(tree.root, 70);

tree.insert(tree.root, 20);

tree.insert(tree.root, 40);

tree.insert(tree.root, 60);

tree.insert(tree.root, 80);

tree.insert(tree.root, 12);

int val;

cout << "Enter value to find: ";

cin >> val;

bool check = tree.find(tree.root, val);

if (check)

cout << val << " is found in BST " << endl;

else

cout << val << " is not found in BST " << endl;

cout << endl;

system("pause");

return 0;

}

## **Screenshot**

