**Name: Areeb Ahmed**

**Reg.No: FA19-BSE-022**

**Section: B**

**Lab Final: DSA**

**Question 1:**

**Code:**

#include <iostream>

using namespace std;

struct node

{

int info;

node \*left,\*right;

};

class bin\_search\_tree

{

private :

node \*temp;

public :

node \*root;

int number;

bin\_search\_tree();

void options();

void b\_search\_tree(node \*);

void in\_order(node \*);

};

int main()

{

char ch;

bin\_search\_tree obj;

while( 4 )

{

obj.options();

cin >> ch;

switch(ch)

{

case '1':

cout<<"\n Enter number to add in a tree... \n";

cin>>obj.number;

obj.b\_search\_tree( obj.root );

break;

case '2':

obj.in\_order(obj.root);

break;

case '3':

exit(0);

break;

default :

exit(0);

break;

}

}

}

bin\_search\_tree :: bin\_search\_tree()

{

root=temp=NULL;

}

void bin\_search\_tree :: b\_search\_tree(node \*temp)

{

if( root==NULL )

{

temp=new node;

temp->info = number;

temp->left=NULL;

temp->right=NULL;

root=temp;

return;

}

if( temp->info==number )

{

cout<<" \n Given number is already present in tree.\n";

return;

}

if(temp->info > number)

{

if( temp->left!=NULL )

{

b\_search\_tree(temp->left);

return;

}

else

{

temp->left=new node;

temp->left->info = number;

temp->left->left=NULL;

temp->left->right=NULL;

return;

}

}

if(temp->info < number)

{

if( temp->right!=NULL )

{

b\_search\_tree( temp->right );

return;

}

else

{

temp->right=new node;

temp->right->info = number;

temp->right->left=NULL;

temp->right->right=NULL;

return;

}

}

}

void bin\_search\_tree :: options()

{

cout<<"\n\n \*\*\*\*\*\*\*\*\*\*\*\*\*\* Select Option \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.\n";

cout<<"\n Enter any of choices.\n";

cout<<"\n 1 : Adding (inserting) node in BST.\n";

cout<<"\n 2 : Print the whole BST .\n";

cout<<"\n 3 : Quitting the Program.\n";

}

void bin\_search\_tree :: in\_order(node \*temp)

{

if(root==NULL)

{

cout<<" Tree is empty.\n";

return;

}

if( temp->left!=NULL )

in\_order(temp->left);

cout<<temp->info<<" ";

if( temp->right!=NULL )

in\_order(temp->right);

return;

}

**Question 2:**

**Code:**

#include <bits/stdc++.h>

#include<iostream>

#include <conio.h>

using namespace std;

int findVal(int array[], int number)

{

int inc\_counter= 1, dec\_counter = 1, temp;

for (int i = 1; i < number; i++)

{

if (array[i] < array[i - 1])

{

if (inc\_counter == 1)

dec\_counter++;

else

return -1;

}

else if (array[i] > array[i - 1])

{

if (inc\_counter == 1)

temp = array[i - 1];

if (dec\_counter >= 2)

inc\_counter++;

else

return -1;

}

else if (array[i] == array[i - 1])

return -1;

}

if (inc\_counter >= 2 && dec\_counter >= 2)

return temp;

else

return -1;

}

int main()

{

int array[] = {9,8,7,6,7,8};

int number = sizeof(array) / sizeof(array[0]);

int val = findVal(array, number);

if (val == -1)

cout << "Element not found";

else

cout<<"Value is ";

cout << val;

return 0;

}

**Question 3:**

**Code:**

#include <bits/stdc++.h>

#include<iostream>

#include <conio.h>

using namespace std;

struct node

{

int value;

struct node\* next;

};

void midValue(struct node\* head)

{

int temp = 0;

struct node\* mean = head;

while (head != NULL)

{

if (temp & 1)

mean = mean->next;

++temp;

head = head->next;

}

if (mean != NULL)

printf("The middle element is [%d]\n\n",

mean->value);

}

void push(struct node\*\* ptr\_head, int new\_val)

{

struct node\* new\_node = (struct node\*)malloc(

sizeof(struct node));

new\_node->value = new\_val;

new\_node->next = (\*ptr\_head);

(\*ptr\_head) = new\_node;

}

void printList(struct node\* ptr)

{

while (ptr != NULL)

{

printf("%d ->", ptr->value);

ptr = ptr->next;

}

printf("NULL\n");

}

int main()

{

struct node\* head = NULL;

int i;

for(i = 5; i > 0; i--)

{

push(&head, i);

printList(head);

midValue(head);

}

return 0;

}