// insertion / deletion in arrays

namespace Day3\_Demos

{

internal class Program

{

static int[] num = new int[10] { 2, 4, 5, 10, 12, 20, 0,0,0,0 };

static void Main(string[] args)

{

PrintElements();

InsertElement(1);

//PrintElements();

InsertElement(30);

//PrintElements();

InsertElement(6);

PrintElements();

}

static void PrintElements()

{

for(int i= 0; i<num.Length; i++)

{

Console.WriteLine(num[i]);

}

}

static int InsertElement(int x)

{

int pos=0;

for (int i = 0; i < num.Length; i++)

{

if (num[i] == 0)

{

pos = i;

break;

}

}

Console.WriteLine("Pos " + pos);

if (x <= num[0])

{

for (int i = pos - 1; i >= 0; i--)

{

num[i + 1] = num[i];

}

num[0] = x;

}

else if (x >= num[pos - 1])

{

num[pos] = x;

}

else

{

for (int i = 0; i < num.Length; i++)

{

if (x >= num[i] && x < num[i + 1])

{

for (int j = pos - 1; j > i; j--)

{

num[j + 1] = num[j];

}

num[i] = x;

break;

}

}

}

return pos;

}

}

}

// insertion / deletion in arrays

namespace Day3\_Demos

{

internal class Program

{

static int[] num = new int[10] { 2, 4, 5, 10, 12, 20, 0, 0, 0, 0 };

static void Main(string[] args)

{

string choice = "y";

while (choice == "y")

{

int ch = Menu();

if (ch == 1)

{

Console.WriteLine("Enter Element to insert in stack");

int num = Int16.Parse(Console.ReadLine());

InsertElement(num);

}

//else if (ch == 2)

//{

// DeleteElement();

//}

else if (ch == 3)

{

PrintElements();

}

else

{

Console.WriteLine("Invalid choice");

}

Console.WriteLine("Repeat again");

choice = Console.ReadLine();

}

}

static int Menu()

{

Console.WriteLine("1. Insert ELement");

Console.WriteLine("2. Delete Element");

Console.WriteLine("3. Display Elements");

Console.WriteLine("Enter your choice");

int ch = int.Parse(Console.ReadLine());

return ch;

}

static void PrintElements()

{

for (int i = 0; i < num.Length; i++)

{

Console.WriteLine(num[i]);

}

}

static int InsertElement(int x)

{

int pos = 0;

for (int i = 0; i < num.Length; i++)

{

if (num[i] == 0)

{

pos = i;

break;

}

}

Console.WriteLine("Pos " + pos);

if (x <= num[0])

{

for (int i = pos - 1; i >= 0; i--)

{

num[i + 1] = num[i];

}

num[0] = x;

}

else if (x >= num[pos - 1])

{

num[pos] = x;

}

else

{

for (int i = 0; i < num.Length; i++)

{

if (x >= num[i] && x < num[i + 1])

{

for (int j = pos - 1; j > i; j--)

{

num[j + 1] = num[j];

}

num[i] = x;

break;

}

}

}

return pos;

}

}

}

-------------------------------------------------------------------------------

Stack

// insertion / deletion in arrays

namespace Day3\_Demos

{

internal class Program

{

static int[] stack = new int[10];

static int top = -1;

static void Main(string[] args)

{

string choice = "y";

while (choice == "y")

{

int ch = Menu();

if (ch == 1)

{

Console.WriteLine("Enter Element to insert in stack");

int num = Int16.Parse(Console.ReadLine());

Push(num);

}

else if (ch == 2)

{

Pop();

}

else if (ch == 3)

{

PrintElements();

}

else

{

Console.WriteLine("Invalid choice");

}

Console.WriteLine("Repeat again");

choice = Console.ReadLine();

}

}

static int Menu()

{

Console.WriteLine("1. Push");

Console.WriteLine("2. Pop");

Console.WriteLine("3. Display Elements");

Console.WriteLine("Enter your choice");

int ch = int.Parse(Console.ReadLine());

return ch;

}

static int Push(int num)

{

if (top > stack.Length)

Console.WriteLine("Overflow");

stack[++top] = num;

return top;

}

static int Pop()

{

return --top;

}

static void PrintElements()

{

for(int i = top; i >= 0; i--)

{

Console.WriteLine(i);

}

}

}

}

// insertion / deletion in arrays

namespace Day3\_Demos

{

internal class Program

{

static int[] queue = new int[10] ;

static int front = -1, rear = -1;

static void Main(string[] args)

{

string choice = "y";

while (choice == "y")

{

int ch = Menu();

if (ch == 1)

{

Console.WriteLine("Enter Element to insert in stack");

int num = Int16.Parse(Console.ReadLine());

Console.WriteLine(InsertElement(num));

}

//else if (ch == 2)

//{

// DeleteElement();

//}

else if (ch == 3)

{

PrintElements();

}

else

{

Console.WriteLine("Invalid choice");

}

Console.WriteLine("Repeat again");

choice = Console.ReadLine();

}

}

static int Menu()

{

Console.WriteLine("1. Insert ELement");

Console.WriteLine("2. Delete Element");

Console.WriteLine("3. Display Elements");

Console.WriteLine("Enter your choice");

int ch = int.Parse(Console.ReadLine());

return ch;

}

static void PrintElements()

{

for (int i = front; i < rear; i++)

{

Console.WriteLine(queue[i]);

}

}

static int InsertElement(int x)

{

if (rear == queue.Length)

Console.WriteLine("Overflow");

else if (front == rear && front == -1)

{ rear = front =0;

queue[front] = x;

}

else

{

Console.WriteLine("rear " + rear);

queue[++rear] = x;

}

return rear;

}

}

}

Single Linked List

using System;

class Node

{

public int data;

public Node rptr;

public Node()

{

data = 0;

rptr = null;

}

}

class Program

{

static Node start = null, last = null, new1, ptr=null, prev=null;

static void Main()

{

string choice = "y";

while(choice == "y") {

int ch = Menu();

switch(ch)

{

case 1:

{

Console.WriteLine("Enter Data to insert");

int data = int.Parse(Console.ReadLine());

Insert(data);

break;

}

case 3:

{

DisplayData();

break;

}

}

Console.WriteLine("Repeat ?");

choice = Console.ReadLine();

}

}

static int Menu()

{

Console.WriteLine("1. Insert ELement");

Console.WriteLine("2. Delete Element");

Console.WriteLine("3. Display Elements");

Console.WriteLine("Enter your choice");

int ch = int.Parse(Console.ReadLine());

return ch;

}

static void Insert(int data)

{

new1 = new Node();

new1.data = data;

new1.rptr = null;

if (start == null)

{

last = start = new1;

Console.WriteLine("Start Node has been added");

}

else if (new1.data >= last.data)

{

last.data = new1.data;

last.rptr = new1;

new1.rptr = null;

last = new1;

Console.WriteLine("Last node added");

}

else

{

Console.WriteLine("New Data " + new1.data);

for (prev = ptr = start; ptr.rptr != null; prev = ptr, ptr = ptr.rptr)

{

Console.WriteLine("Current Data " + ptr.data);

Console.WriteLine("Prev Data " + prev.data);

if (new1.data >= prev.data && new1.data < ptr.data)

{

prev.rptr = new1;

new1.rptr = ptr;

break;

}

}

}

}

static void DisplayData()

{

for(ptr=start; ptr!=null; ptr=ptr.rptr)

{

Console.WriteLine(ptr.data);

}

}

}

BST

using System;

class Node

{

public int data;

public Node lptr;

public Node rptr;

public void Dispaly()

{

Console.WriteLine(data + "---");

}

}

class BST

{

public Node root;

public BST()

{

root = null;

}

public void InsertData(int data)

{

Node newNode = new Node();

newNode.data = data;

if(root == null)

{

root = new Node();

root = newNode;

Console.WriteLine("Root Node has been added");

}

else

{

Node ptr = root;

Node parent;

while(true)

{

parent = ptr;

if (data < ptr.data)

{

ptr = ptr.lptr;

if (ptr == null)

{

parent.lptr = newNode;

Console.WriteLine("Added at left");

break;

}

} else

{

ptr = ptr.rptr;

if(ptr==null)

{

parent.rptr = newNode;

Console.WriteLine("Added at right");

break;

}

}

}

}

}

public void PreOrder(Node parent)

{

if (parent != null)

{

Console.WriteLine(parent.data);

PreOrder(parent.lptr);

PreOrder(parent.rptr);

}

}

public void InOrder(Node parent)

{

if (parent != null)

{

InOrder(parent.lptr);

Console.WriteLine(parent.data);

InOrder(parent.rptr);

}

}

public void PostOrder(Node parent)

{

if(parent!=null)

{

PostOrder(parent.lptr);

PostOrder(parent.rptr);

Console.WriteLine(parent.data);

}

}

}

class Program

{

static void Main()

{

string choice = "y";

BST bST = new BST();

bST.InsertData(10);

bST.InsertData(12);

bST.InsertData(15);

bST.InsertData(5);

bST.InsertData(7);

Console.WriteLine("Preorder ------ ");

bST.PreOrder(bST.root);

Console.WriteLine("Postorder ------ ");

bST.PostOrder(bST.root);

Console.WriteLine("Inorder ------ ");

bST.InOrder(bST.root);

}

static int Menu()

{

Console.WriteLine("1. Insert ELement");

Console.WriteLine("2. Delete Element");

Console.WriteLine("3. Display Elements");

Console.WriteLine("Enter your choice");

int ch = int.Parse(Console.ReadLine());

return ch;

}

}