public void RemoveAt(int i)

{

if (i < 0 || i >= Count)

{

throw new IndexOutOfRangeException();

}

if (i == 0)

{

head = head.Next;

if (head == null)

{

tail = null;

}

}

else

{

Node current = head;

for (int j = 0; j < i - 1; j++)

{

current = current.Next;

}

current.Next = current.Next.Next;

if (current.Next == null)

{

tail = current;

}

}

Count--;

}

public void RemoveX(int x)

{

Node current = head;

Node previous = null;

while (current != null)

{

if (current.Data == x)

{

if (previous == null)

{

head = current.Next;

if (head == null)

{

tail = null;

}

}

else

{

previous.Next = current.Next;

if (current.Next == null)

{

tail = previous;

}

}

Count--;

return;

}

previous = current;

current = current.Next;

}

}

public void InsertAt(int x, int i)

{

if (i < 0 || i > Count)

{

throw new IndexOutOfRangeException();

}

Node newNode = new Node { Data = x };

if (i == 0)

{

newNode.Next = head;

head = newNode;

if (tail == null)

{

tail = newNode;

}

}

else

{

Node current = head;

for (int j = 0; j < i - 1; j++)

{

current = current.Next;

}

newNode.Next = current.Next;

current.Next = newNode;

if (newNode.Next == null)

{

tail = newNode;

}

}

Count++;

}

public void InsertXAfterMin(int x)

{

Node minNode = GetMin();

Node newNode = new Node { Data = x };

newNode.Next = minNode.Next;

minNode.Next = newNode;

if (newNode.Next == null)

{

tail = newNode;

}

Count++;

}

public void InsertXAfterY(int x, int y)

{

Node yNode = SearchX(y);

if (yNode == null)

{

throw new ArgumentException("Value y not found in the list.");

}

Node newNode = new Node { Data = x };

}

public void InsertXBeforeMax(int x)

{

Node maxNode = GetMax();

Node newNode = new Node { Data = x };

if (head == maxNode)

{

newNode.Next = head;

head = newNode;

}

else

{

Node current = head;

while (current.Next != maxNode)

{

current = current.Next;

}

newNode.Next = current.Next;

current.Next = newNode;

}

Count++;

}

public void InsertXBeforeY(int x, int y)

{

Node yNode = SearchX(y);

Node newNode = new Node { Data = x };

if (head == yNode)

{

newNode.Next = head;

head = newNode;

}

else

{

Node current = head;

while (current.Next != yNode)

{

current = current.Next;

}

newNode.Next = current.Next;

current.Next = newNode;

}

Count++;

}

public void RShiftRight()

{

if (head == null || head == tail)

{

return;

}

Node current = head;

while (current.Next != tail)

{

current = current.Next;

}

tail.Next = head;

head = tail;

tail = current;

tail.Next = null;

}

public void InterchangeSort()

{

for (Node i = head; i != null; i = i.Next)

{

for (Node j = i.Next; j != null; j = j.Next)

{

if (i.Data > j.Data)

{

int temp = i.Data;

i.Data = j.Data;

j.Data = temp;

}

}

}

}

public void SelectionSort()

{

for (Node i = head; i != null; i = i.Next)

{

Node maxNode = i;

for (Node j = i.Next; j != null; j = j.Next)

{

if (j.Data > maxNode.Data)

{

maxNode = j;

}

}

if (maxNode != i)

{

int temp = i.Data;

i.Data = maxNode.Data;

maxNode.Data = temp;

}

}

}

static void Main(string[] args)

{

MyList myList = new MyList();

int choice, x, y;

Node node;

do

{

Console.OutputEncoding = Encoding.UTF8;

Console.WriteLine("\nMENU");

Console.WriteLine("1. Nhập các giá trị vào danh sách");

Console.WriteLine("2. Hiển thị danh sách");

Console.WriteLine("3. Xóa node tại vị trí thứ i");

Console.WriteLine("4. Xóa node có giá trị x");

Console.WriteLine("5. Chèn x vào danh sách tại vị trí thứ i");

Console.WriteLine("6. Chèn x vào sau node có giá trị nhỏ nhất");

Console.WriteLine("7. Chèn x vào sau node có giá trị y");

Console.WriteLine("8. Chèn x vào trước node có giá trị lớn nhất");

Console.WriteLine("9. Chèn x vào trước node có giá trị y");

Console.WriteLine("10. Dịch phải xoay vòng danh sách");

Console.WriteLine("11. Sắp xếp danh sách theo thứ tự tăng dần (Interchange sort)");

Console.WriteLine("12. Sắp xếp danh sách theo thứ tự giảm dần (Selection sort)");

Console.WriteLine("23. Thoát");

Console.Write("\nNhập lựa chọn của bạn: ");

choice = Convert.ToInt32(Console.ReadLine());

switch (choice)

{

case 1:

myList.Input();

break;

case 2:

myList.ShowList();

break;

case 3:

Console.Write("\nNhập vị trí để xóa: ");

int i = Convert.ToInt32(Console.ReadLine());

myList.RemoveAt(i);

break;

case 4:

Console.Write("\nNhập giá trị để xóa: ");

x = Convert.ToInt32(Console.ReadLine());

myList.RemoveX(x);

break;

case 5:

Console.Write("\nNhập giá trị để chèn: ");

x = Convert.ToInt32(Console.ReadLine());

Console.Write("\nNhập vị trí để chèn: ");

i = Convert.ToInt32(Console.ReadLine());

myList.InsertAt(x, i);

break;

case 6:

Console.Write("\nNhập giá trị để chèn sau min: ");

x = Convert.ToInt32(Console.ReadLine());

myList.InsertXAfterMin(x);

break;

case 7:

Console.Write("\nNhập giá trị x để chèn: ");

x = Convert.ToInt32(Console.ReadLine());

Console.Write("\nNhập giá trị y để chèn sau: ");

y = Convert.ToInt32(Console.ReadLine());

myList.InsertXAfterY(x, y);

break;

case 8:

Console.Write("\nNhập giá trị để chèn trước max:");

x = Convert.ToInt32(Console.ReadLine());

myList.InsertXBeforeMax(x);

break;

case 9:

Console.Write("\nNhập giá trị x để chèn: ");

x = Convert.ToInt32(Console.ReadLine());

Console.Write("\nNhập giá trị y để chèn trước: ");

y = Convert.ToInt32(Console.ReadLine());

myList.InsertXBeforeY(x, y);

break;

case 10:

myList.RShiftRight();

break;

case 11:

myList.InterchangeSort();

break;

case 12:

myList.SelectionSort();

break;

case 13:

Environment.Exit(0);

break;

default:

Console.WriteLine("\nLựa chọn không hợp lệ. Vui lòng thử lại.");

break;

}

} while (true);

}