Q:1 in c

Code:

#include<stdio.h>

#include<stdlib.h>

struct node{

    int data;

    struct node \*next;

};

struct node \*head=NULL;

void insertAtHead(int data){

    struct node\*ptr=(struct node\*)malloc(sizeof(struct node));

    struct node\*n1=head;

    ptr->next=head;

    head=ptr;

    ptr->data=data;

}

void insertAtTail(int data){

    struct node\*ptr=(struct node\*)malloc(sizeof(struct node));

    struct node\*n1=head;

    if(head==NULL){

        head=ptr;

    }else{

    while(n1->next!=NULL){

        n1=n1->next;

    }

    n1->next=ptr;

    }

    ptr->data=data;

    ptr->next=NULL;

}

void insertAt(int index,int data){

     struct node\*ptr=(struct node\*)malloc(sizeof(struct node));

    struct node\*n1=head;

    int j=0;

    while(j<index-1){

        n1=n1->next;

        j++;

    }

    struct node \*n5=ptr;

    ptr->next=n1->next;

    n1->next=n5;

    ptr->data=data;

}

void delete(int data){

    if(head->data==data){

        head=head->next;

    }else{

     struct node\*n1=head;

     while(n1->next->data!=data){

        n1=n1->next;

     }

     n1->next=n1->next->next;

    }

}

void show(){

    struct node\* n1=head;

while(n1!=NULL){

    printf("%d ",n1->data);

    n1=n1->next;

}

}

int main(){

    insertAtHead(12);

    insertAtHead(10);

    insertAtHead(19);

    insertAtHead(118);

   insertAtHead(117);

    insertAt(2,234);

    insertAtTail(345);

    delete(19);

    show();

}

Output:

117 118 234 10 12 345

Q:2 linklist in java

Code:

import java.util.\*;

class node {

    Object data;

    node next;

    node pre;

}

class arraylist {

    node head;

    public node tail;

    public void add(Object data) {

        node ne = new node();

        if (head == null) {

            head = ne;

            ne.pre = null;

            tail = ne;

        } else {

            node n1 = head;

            while (n1.next != null) {

                n1 = n1.next;

            }

            n1.next = ne;

            ne.pre = n1;

            tail = ne;

        }

        ne.data = data;

        ne.next = null;

    }

    public void rev() {

        node t2 = tail;

        while (t2 != null) {

            System.out.print(t2.data + " ");

            t2 = t2.pre;

        }

    }

    public void remove(int i) {

        if (i == 0) {

            head = head.next;

            head.pre = null;

        } else {

            int j = 0;

            node n1 = head;

            while (j != i - 1) {

                n1 = n1.next;

                j++;

            }

            node n2 = n1.next;

            n1.next = n2.next;

            n2.next.pre = n2.pre;

        }

    }

    public void insertAt(int i, Object data) {

        node ne = new node();

        ne.data = data;

        if (i == 0) {

            ne.next = head;

            head = ne;

        } else {

            int j = 0;

            node n1 = head;

            while (j != i - 1) {

                n1 = n1.next;

            }

            ne.next = n1.next;

            n1.next = ne;

        }

    }

    public Object get(int index) {

        node n1 = head;

        if (index == 0) {

            return n1.data;

        } else {

            int i = 0;

            while (i != index) {

                n1 = n1.next;

                i++;

            }

        }

        return n1.data;

    }

    public int size() {

        node n1 = head;

        int j = 0;

        while (n1 != null) {

            n1 = n1.next;

            j++;

        }

        return j;

    }

    public boolean contain(Object data) {

        node n1 = head;

        while (n1.next != null) {

            if (n1.data == data) {

                return true;

            }

            n1 = n1.next;

        }

        return false;

    }

    node recursive(node head) {

        if (head == null || head.next == null)

            return head;

        node newhead = recursive(head.next);

        head.next.next = head;

        head.next = null;

        return newhead;

    }

    void show() {

        node n2 = head;

        System.out.print("[ ");

        while (n2 != null) {

            System.out.print(n2.data + " ");

            n2 = n2.next;

        }

        System.out.print("]\n");

    }

}

public class linkliste {

    public static void main(String[] args) {

        arraylist a1 = new arraylist();

        a1.add("sabalpara");

        a1.add("jay");

        a1.add(293);

        a1.add("@");

        a1.add("gmail.com");

        a1.add(12);

        a1.add(111);

        a1.add(3);

        a1.add(4);

        a1.add(5);

        a1.remove(2);

        a1.insertAt(1, "hi");

        System.out.println(a1.get(3));

        System.out.println(a1.size());

        System.out.println(a1.contain("hii"));

        a1.show();

        a1.head = a1.recursive(a1.head);

        a1.show();

    }

}

Output:

@

10

false

[ sabalpara hi jay @ gmail.com 12 111 3 4 5 ]

[ 5 4 3 111 12 gmail.com @ jay hi sabalpara ]

PS C:\java>

Q:3 >25 numbers

Code:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                            Online Java Compiler.

                Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

import java.util.\*;

class node {

    Object data;

    node next;

}

class arraylist {

    node head;

    public void add(Object data) {

        node ne = new node();

        if (head == null) {

            head = ne;

        } else {

            node n1 = head;

            while (n1.next != null) {

                n1 = n1.next;

            }

            n1.next = ne;

        }

        ne.data = data;

        ne.next = null;

    }

    public int size() {

        node n1 = head;

        int j = 0;

        while (n1 != null) {

            n1 = n1.next;

            j++;

        }

        return j;

    }

    void tfive() {

        // node t2=head;

        if ((int) head.data > 25) {

            head = head.next;

        }

        node t1 = head;

        while (t1 != null) {

            if (t1.next == null) {

                if ((int) t1.data > 25) {

                    t1 = t1.next;

                    continue;

                }

            } /\*

               \* else if(t1.next.next==null)

               \* {

               \* if((int)t1.next>25){

               \*

               \* }

               \* }

               \*/

            else if ((int) t1.next.data > 25) {

                t1.next = t1.next.next;

                continue;

            }

            t1 = t1.next;

        }

    }

    void show() {

        node n2 = head;

        System.out.print("[ ");

        while (n2 != null) {

            System.out.print(n2.data + " ");

            n2 = n2.next;

        }

        System.out.print("]\n");

    }

}

public class linklistee {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        arraylist a1 = new arraylist();

        int[] arr = new int[100];

        for (int i = 0; i < 100; i++) {

            arr[i] = sc.nextInt();

        }

        for (int i = 0; i < 100; i++) {

            a1.add(arr[i]);

        }

        a1.tfive();

        a1.show();

    }

}

Output:

1 2 3 4 5 6 7 8 9 0 11 12 13 14 15 16 17 18 19 20 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 62 61 63 64 65 66 67 6

8 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 100 21 22 23 24 25 26 27 28 29 30

12

[ 1 2 3 4 5 6 7 8 9 0 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 12 ]

PS C:\java>

Q:4

import java.util.\*;

class node {

    int data;

    node next;

}

class arraylist {

    node head;

    public void add(int data) {

        node ne = new node();

        if (head == null) {

            head = ne;

        } else {

            node n1 = head;

            while (n1.next != null) {

                n1 = n1.next;

            }

            n1.next = ne;

        }

        ne.data = data;

        ne.next = null;

    }

    public int contain(int data) {

        node n1 = head;

        int j = 0;

        while (n1.next != null) {

            if (n1.data == data) {

                return j;

            }

            n1 = n1.next;

            j++;

        }

        return -1;

    }

}

public class llindex {

    public static void main(String[] args) {

        arraylist a1 = new arraylist();

        a1.add(23443);

        a1.add(46);

        a1.add(293);

        a1.add(67);

        int n = a1.contain(46);

        System.out.println(n);

    }

}

Output:

1