|  |
| --- |
| Código Original  int opc=1; // La bandera que permita capturar datos  Scanner sc = new Scanner(System.in); //Se creo un objeto tipo escanner.    Lista P = new Lista(); // Esta es la lista que vamos a utilizar  Lista Q = new Lista(); // No esta siendo utilizada  P.primero = null; //Esta lista se encuentra vacia  Q.primero = null; //Esta lista se encuentra vacia.    //Permite crear un primer NODO  nodo r = new nodo(); // Puntero temporal para saltar de nodo a nodo  nodo n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Primer Nodo");  n.info = sc.nextInt();  n.siguiente = null;  P.primero = n;  r=P.primero;    //De aqui en eadelante los que deseemos - NODOS  while (opc != 0)  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = null;  r.siguiente = n;  r= r.siguiente;  System.out.println("Desea crear otro Nodo 1-Si, 0-NO");  opc = sc.nextInt(); // Si es 1 - Siguie Pidiendo y si es = se sale  }  P.imprimirLista();    // Deseamos eliminar cualquier NOdo  nodo s = new nodo();  s=r=P.primero;  int c=1,cualNodo=0;  System.out.print("Cual Nodo Eliminar ");  cualNodo = sc.nextInt();  while(r.siguiente!=null){  //Los nodos que voy saltando  ++c;  s=r;  r=r.siguiente;  if(cualNodo == 1){  P.primero=r;  break;  }else  if(cualNodo == c)  {  s.siguiente = r.siguiente;  }  }  P.imprimirLista();  //Sumar el primero con el último  int su=0;  r = P.primero;  su+=P.primero.info;  while(r.siguiente != null){  r = r.siguiente;  }  su+=r.info;  System.out.println("La suma del Primero con el último fue de: " +su);  // LA SUMA DEL PRIMERO CON EL ÚLTIMO - Otra versión  int bandera = 0,suma=0;  r=P.primero;  while(r.siguiente != null){  if(bandera==0)  {  suma=r.info;  bandera=1;  }    r=r.siguiente;  }  suma += r.info;  System.out.println("La Suma del Primero más el último es " + suma);  // Determinar la Suma de los Campos de iNformación PAR  int sp=0;  r=P.primero;  while (r.siguiente != null){  if (r.info % 2 == 0)  sp += r.info;  r=r.siguiente;  }  System.out.println("La suma de los pares es de " + sp); |
| /\* El primero nodo más el último número impar de la lista. \*/  r=P.primero;  /\*La suma del primero más el último Impar\*/  int ban1=0,ElPrimero=0,UltimoImpar=0;  while(r!=null){  if(ban1==0)  {  ElPrimero = r.info;  ban1 =1;  }else{  if(r.info % 2 != 0)  UltimoImpar = r.info;  }  r = r.siguiente;  }  if(UltimoImpar == 0)  if(ElPrimero % 2 != 0)  UltimoImpar = ElPrimero;  System.out.println("La suma del Primero " + ElPrimero + " Mas el último impar es : " +  UltimoImpar + " es " + (ElPrimero + UltimoImpar)); |
| // El Primer Par más el último par  int PrimerPar,UltimoPar;  PrimerPar = UltimoPar =0;  int ban1=0,suma=0;  r=P.primero;  while(r!=null){  if(r.info % 2 == 0 && ban1 ==0)  {  PrimerPar = r.info;  ban1 = 1;  }  else{  if(r.info % 2 == 0)  {  UltimoPar = r.info;  }  }  r = r.siguiente;  }  suma = PrimerPar +UltimoPar;  System.out.println("La suma es " + suma); |
| /\*  Lineales Circulares  \*/  package listalinealescirculares;  import java.util.Scanner;  public class ListaLinealesCirculares {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in); //Se creo un objeto tipo escanner.  int bandera = 0,bandera1=0;  int opc =1, opc1 = 1;  Lista P = new Lista(); // Esta es la lista que vamos a utilizar  Lista Q = new Lista(); // No esta siendo utilizada  P.primero = null; //Esta lista se encuentra vacia  Q.primero = null; //Esta lista se encuentra vacia.    //Permite crear un primer NODO  nodo r = new nodo(); // Puntero temporal para saltar de nodo a nodo  nodo n = new nodo(); // Creamos un nuevo nodo    //De aqui en eadelante los que deseemos - NODOS  while (opc != 0)  {  if(bandera == 0)  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  bandera = 1; // Nunca más vuelva a ingresar aquí.  P.primero = n;  r=P.primero;  n.siguiente = P.primero;  }else  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = P.primero;  r.siguiente = n;  r= r.siguiente;  }  System.out.println("Desea crear otro Nodo 1-Si, 0-NO");    }  // Segunda Lista Circular  //De aqui en eadelante los que deseemos - NODOS  System.out.println("NODOS DE LA SEGUNDA LISTA");  nodo s = new nodo();  while (opc1 != 0)  {  if(bandera1 == 0)  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  bandera1 = 1; // Nunca más vuelva a ingresar aquí.  Q.primero = n;  s=Q.primero;  n.siguiente = Q.primero;  }else  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = Q.primero;  s.siguiente = n;  s= s.siguiente;  }  System.out.println("Desea crear otro Nodo 1-Si, 0-NO");  opc1 = sc.nextInt(); // Si es 1 - Siguie Pidiendo y si es = se sale  }    Q.imprimirLista();      }  public static class Lista{  nodo primero;  //Se encapsulo el codigo Imprimir Lista Como un Metodo.  public void imprimirLista(){  nodo r = new nodo();  r=primero;  while(r.siguiente!=primero){  System.out.println("El Nodo es " + r.info);  r = r.siguiente;  }  System.out.println("El Nodo es " + r.info);  }  }  private static class nodo {  int info;  nodo siguiente;  public nodo() {  }  }  } |
| /\*  Lineales Circulares Horario 8 pm Listas Lineales circulares la concatenación de dos.  \*/  package listalinealescirculares;  import java.util.Scanner;  public class ListaLinealesCirculares {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in); //Se creo un objeto tipo escanner.  int bandera = 0,bandera1=0;  int opc =1, opc1 = 1;  Lista P = new Lista(); // Esta es la lista que vamos a utilizar  Lista Q = new Lista(); // No esta siendo utilizada  P.primero = null; //Esta lista se encuentra vacia  Q.primero = null; //Esta lista se encuentra vacia.    //Permite crear un primer NODO  nodo r = new nodo(); // Puntero temporal para saltar de nodo a nodo  nodo s = new nodo(); //Creado para la segunda lista  nodo n = new nodo(); // Creamos un nuevo nodo    //De aqui en eadelante los que deseemos - NODOS  while (opc != 0)  {  if(bandera == 0)  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  bandera = 1; // Nunca más vuelva a ingresar aquí.  P.primero = n;  r=P.primero;  n.siguiente = P.primero;  }else  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = P.primero;  r.siguiente = n;  r= r.siguiente;  }  System.out.println("Desea crear otro Nodo 1-Si, 0-NO");  opc = sc.nextInt();  }  P.imprimirLista();  // A partir de aquie va ser la lista Q  while (opc1 != 0)  {  if(bandera1 == 0)  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  bandera1 = 1; // Nunca más vuelva a ingresar aquí.  Q.primero = n;  s=Q.primero;  n.siguiente = Q.primero;  }else  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = Q.primero;  s.siguiente = n;  s= s.siguiente;  }  System.out.println("Desea crear otro Nodo 1-Si, 0-NO");  opc1 = sc.nextInt();  }  Q.imprimirLista();  System.out.println("---------------------------------------------");  System.out.println("---------------------------------------------");  //De aqui en adelante concateno las listas  r=P.primero;  while(r.siguiente != P.primero){  r=r.siguiente;  }  r.siguiente = Q.primero;  r = r.siguiente;  while(r.siguiente != Q.primero)  {  r = r.siguiente;  }  r.siguiente = P.primero;  P.imprimirLista();  System.out.println("---------------------------------------------");  System.out.println("---------------------------------------------");  Q.imprimirLista();    }  public static class Lista{  nodo primero;  //Se encapsulo el codigo Imprimir Lista Como un Metodo.  public void imprimirLista(){  nodo r = new nodo();  r=primero;  while(r.siguiente!=primero){  System.out.println("El Nodo es " + r.info);  r = r.siguiente;  }  System.out.println("El Nodo es " + r.info);  }  }  private static class nodo {  int info;  nodo siguiente;  public nodo() {  }  }  } |
| /\* Dada una llE determinar la suma del primer número PAR encontrado en la lista más la suma del último número primo encontrado en ella.\*/  package listaslineales\_corregido;  import java.util.Scanner;  public class ListasLineales\_corregido {  public static void main(String[] args) {  int opc=1; // La bandera que permita capturar datos  int bandera = 0;  Scanner sc = new Scanner(System.in); //Se creo un objeto tipo escanner.    Lista P = new Lista(); // Esta es la lista que vamos a utilizar  Lista Q = new Lista(); // No esta siendo utilizada  P.Primero = null; //Esta lista se encuentra vacia  Q.Primero = null; //Esta lista se encuentra vacia.    //Permite crear un primer NODO  nodo r = new nodo(); // Puntero temporal para saltar de nodo a nodo  nodo n = new nodo(); // Creamos un nuevo nodo    //De aqui en eadelante los que deseemos - NODOS  while (opc != 0)  {  if(bandera==0)  {  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = null;  r.siguiente = n;  P.Primero=n;  r= P.Primero;  bandera =1;  }else{  r=n;  n = new nodo(); // Creamos un nuevo nodo  System.out.println("Digite el valor del Nodo");  n.info = sc.nextInt();  n.siguiente = null;  r.siguiente = n;  r= r.siguiente;    }  System.out.println("Desea crear otro Nodo 1-Si, 0-NO");  opc = sc.nextInt(); // Si es 1 - Siguie Pidiendo y si es = se sale  }  P.imprimirLista();  // Vamos a encontrar el Primer Par  int PrimerPar = 0,ban2=0,ultimoPrimo=0,cont=0,suma=0;  r=P.Primero;  while(r!=null){  if(r.info %2 == 0 && ban2==0){  PrimerPar = r.info;  ban2 =1;  }  for(int i=1;i<=r.info;i++){  if(r.info % i == 0){  ++cont;  }  }  if(cont <= 2){  ultimoPrimo = r.info;  }    cont=0;  r = r.siguiente;  }  suma = PrimerPar + ultimoPrimo;  System.out.println("La suma del Primer Par mas el último primo es : " + suma);    }  public static class Lista{  nodo Primero;  public void imprimirLista()  {  nodo r = new nodo();  r=Primero;  while(r!=null)  {  System.out.println(r.info);  r = r.siguiente;  }  }  }  public static class nodo{  int info;  nodo siguiente;  }  } |
| /\* Dadas dos listas lineales Circulares, crear una tercera lista linear circular que solo contenga los números pares contenidos en las listas.\*/ |
|  |