import java.util.Calendar;  
import java.util.Scanner;  
class SocialNetworkGraph {  
 static class DOB {  
 int day, month, year;  
 DOB(int day, int month, int year) {  
 this.day = day;  
 this.month = month;  
 this.year = year;  
 }  
 }  
 static class Node {  
 Link e;  
 Node next;  
 String name;  
 int comments;  
 DOB dob;  
 boolean visited;  
 Node(String name, int comments, DOB dob) {  
 this.name = name;  
 this.comments = comments;  
 this.dob = dob;  
 this.visited = false;  
 }  
 }  
 static class Link {  
 Link next;  
 Node ptr;  
 Link(Node p) {  
 this.ptr = p;  
 this.next = null;  
 }  
 }  
 private Node head, insert;  
 void create() {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter number of nodes: ");  
 int n = scanner.nextInt();  
 for (int i = 0; i < n; i++) {  
 System.*out*.println("Enter details of person " + (i + 1));  
 System.*out*.print("Enter name: ");  
 String name = scanner.next();  
 System.*out*.print("Enter date of birth (day month year): ");  
 int day = scanner.nextInt();  
 int month = scanner.nextInt();  
 int year = scanner.nextInt();  
 DOB dob = new DOB(day, month, year);  
 System.*out*.print("Enter number of comments: ");  
 int comments = scanner.nextInt();  
 if (i == 0)  
 head = insert = new Node(name, comments, dob);  
 else {  
 insert.next = new Node(name, comments, dob);  
 insert = insert.next;  
 }  
 }  
 for (Node i = head; i != null; i = i.next) {  
 System.*out*.println("Who are friends of " + i.name + "?");  
 for (Node j = head; j != null; j = j.next) {  
 if (j == i)  
 continue;  
 System.*out*.print("Is " + j.name + " a friend? (y/n): ");  
 char c = scanner.next().charAt(0);  
 if (c == 'y') {  
 Link temp;  
 if (i.e == null) {  
 i.e = new Link(j);  
 continue;  
 }  
 for (temp = i.e; temp.next != null; temp = temp.next);  
 temp.next = new Link(j);  
 }  
 }  
 }  
 }  
 void display() {  
 for (Node i = head; i != null; i = i.next) {  
 System.*out*.print("\nName: " + i.name + " DOB: " + i.dob.day + "/" + i.dob.month + "/" +  
 i.dob.year + " Comments: " + i.comments + "\nFriends: ");  
 for (Link temp = i.e; temp != null; temp = temp.next)  
 System.*out*.print(temp.ptr.name + " ");  
 }  
 }  
 void friends() {  
 int min = Integer.*MAX\_VALUE*, max = 0;  
 Node[] S = new Node[30];  
 int top = 0;  
 S[top] = head;  
 head.visited = true;  
 while (top > -1) {  
 Node temp = S[top--];  
 int n = 0;  
 for (Link l = temp.e; l != null; l = l.next) {  
 n++;  
 if (!l.ptr.visited) {  
 S[++top] = l.ptr;  
 l.ptr.visited = true;  
 }  
 }  
 if (max < n) max = n;  
 if (min > n) min = n;  
 }  
 System.*out*.println("Maximum: " + max + " Minimum: " + min);  
 }  
 void comments() {  
 int min = Integer.*MAX\_VALUE*, max = 0;  
 Node[] Q = new Node[30];  
 int f = -1, r = 0;  
 Q[r++] = head;  
 head.visited = true;  
 while (f != r - 1) {  
 Node temp = Q[++f];  
 if (max < temp.comments) max = temp.comments;  
 if (min > temp.comments) min = temp.comments;  
 for (Link l = temp.e; l != null; l = l.next) {  
 if (!l.ptr.visited) {  
 Q[r++] = l.ptr;  
 l.ptr.visited = true;  
 }  
 }  
 }  
 System.*out*.println("Maximum: " + max + " Minimum: " + min);  
 }  
 void resetVisited() {  
 for (Node t = head; t != null; t = t.next)  
 t.visited = false;  
 }  
 void birthdays() {  
 Calendar now = Calendar.*getInstance*();  
 int month = now.get(Calendar.*MONTH*) + 1;  
 boolean flag = false;  
 System.*out*.println("Birthdays in current month:");  
 for (Node i = head; i != null; i = i.next) {  
 if (i.dob.month == month) {  
 System.*out*.println(i.name + " " + i.dob.day + "-" + i.dob.month + "-" + i.dob.year);  
 flag = true;  
 }  
 }  
 if (!flag) System.*out*.println("No birthdays in this month!");  
 }  
 public static void main(String[] args) {  
 SocialNetworkGraph graph = new SocialNetworkGraph();  
 Scanner scanner = new Scanner(System.*in*);  
 int choice = 0;  
 while (choice != 6) {  
 System.*out*.print("\n1.Create \n2.Display \n3.Friends \n4.Comments \n5.Birthdays \n6.Exit:\n ");  
 choice = scanner.nextInt();  
 switch (choice) {  
 case 1:  
 graph.create();  
 break;  
 case 2:  
 graph.display();  
 break;  
 case 3:  
 graph.resetVisited();  
 graph.friends();  
 break;  
 case 4:  
 graph.resetVisited();  
 graph.comments();  
 break;  
 case 5:  
 graph.birthdays();  
 break;  
 }  
 }  
 }  
}