**Nama : Michael Geraldin Wijaya**

**NIM : 2602238021**

**Class : LC95**

**ASSURANCE OF LEARNING**

1. **PSEUDOCODE**

Declare integer N,T,j,i,c,p,q

Input N

Declare integer array mat ukuran N\*N

**If(N>2) then**

**For i:=0 to N**

**For j:=0 to N**

Input mat[i][j]

**End for**

**End for**

Declare integer x

Set c = N/2

**For x:=0 to 1 (loop once, so the break statement can be used)**

**If mat[c][c]==0 then** Display “The value of centroid can’t be 0”

Break

**End if**

Display (centroid at c,c)

Set p = 0

Set q = 0

**If mat[c + 1][c] == mat[c][c] OR mat[c][c + 1] == mat[c][c]**

**OR mat[c][c - 1] == mat[c][c] OR mat[c - 1][c] == mat[c][c] then**

**For i:=c+1 to c-1**

**For j:=c+1 to c-1**

**If mat[i][j]==mat[c][c] then**

**If i==c AND j==c then** continue

**End if**

Set p++

**End if**

**End for**

**End for**

**Else if mat[c + 1][c + 1] == mat[c][c] OR mat[c + 1][c - 1] == mat[c][c]**

**OR mat[c - 1][c + 1] == mat[c][c] OR mat[c - 1][c - 1] == mat[c][c] then**

**For i:=c+1 to c-1**

**For j:=c+1 to c-1**

**If mat[i][j]==mat[c][c] then**

**If i==c AND j==c then** continue

**End if**

Set q++

**End if**

**End for**

**End for**

**End if**

Display “Nearest same elements is at:”

**If p>0 then**

**For i:=c+1 to c-1**

**For j:=c+1 to c-1**

**If mat[i][j]==mat[c][c] then**

**If i==c+1 AND j==c OR i==c AND j==c+1 OR i==c AND j==c-1 OR i==c-1 AND j==c then**

**If i==c AND j==c then** continue

**End if**

Display (i,j)

**End if**

**End if**

**End for**

**End for**

**Else if p==0 then**

**If q==0 then** Display “ no nearest element.”

**Else if q>0 then**

**For i:=c+1 to c-1**

**For j:=c+1 to c-1**

**If mat[i][j]==mat[c][c] then**

**If i==c AND j==c then** continue

**End if**

Display (i,j)

**End if**

**End for**

**End for**

**End if**

**End if**

**End for**

**Else** Display “Size of matrix must be more than 2\*2”

**End if**

**FLOWCHART**

**Diagram

Description automatically generated**

**Diagram, engineering drawing

Description automatically generated**

**Diagram, engineering drawing

Description automatically generated**

1. **PSEUDOCODE**

**Struct** Friend(

    string name

    string phoneNumber

    string address

)as array friends

**Function** searchFriends(string pointer nameee)

Declareinteger i, found

Set found = 0

**For i:=0 to strlen(friends)**

**If strstr(friends[i].name, nameee not NULL** **then**

Display friends[i].name friends[i].phoneNumber friends[i].address

Set found = 1

**End if**

**End for**

**Return** found

**End function**

**Module main()**

Declare integer i

Declare string namee

Declare char stopInput

Set stopInput = Y

Display “Input data: Name, Phone Number, and Address”

Set i = 0

**While** **stopInput == Y**

Display “Name: ”

Input friends[i].name

Display “Phone Number: ”

Input friends[i].phoneNumber

Display “Address: ”

Input friends[i].address

Display “Want to input data again? (Y/N): ”

Input stopInput

Set i++

**End While**

Display “Enter a name to search: ”

Input namee

Declare integer founds

Set founds = searchFriends(namee)

**If founds not 1 then** Display “Data Not Exist”

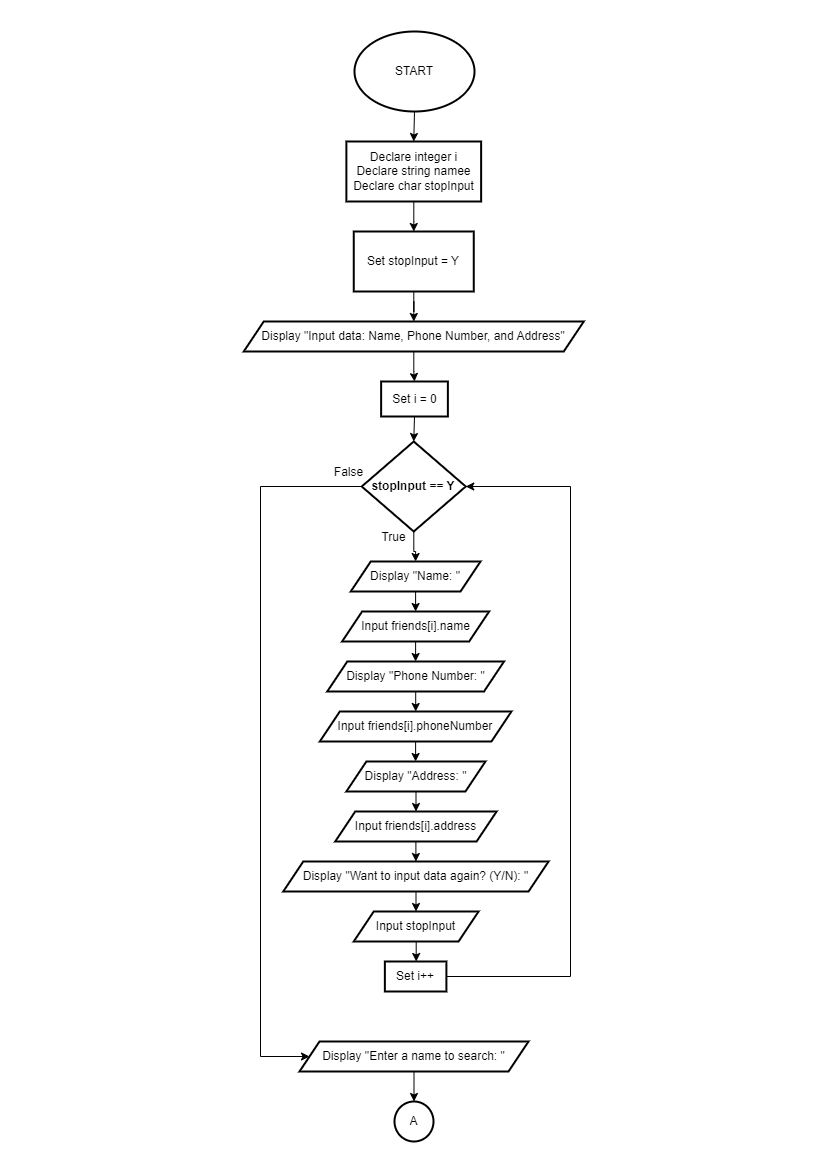
**End if**

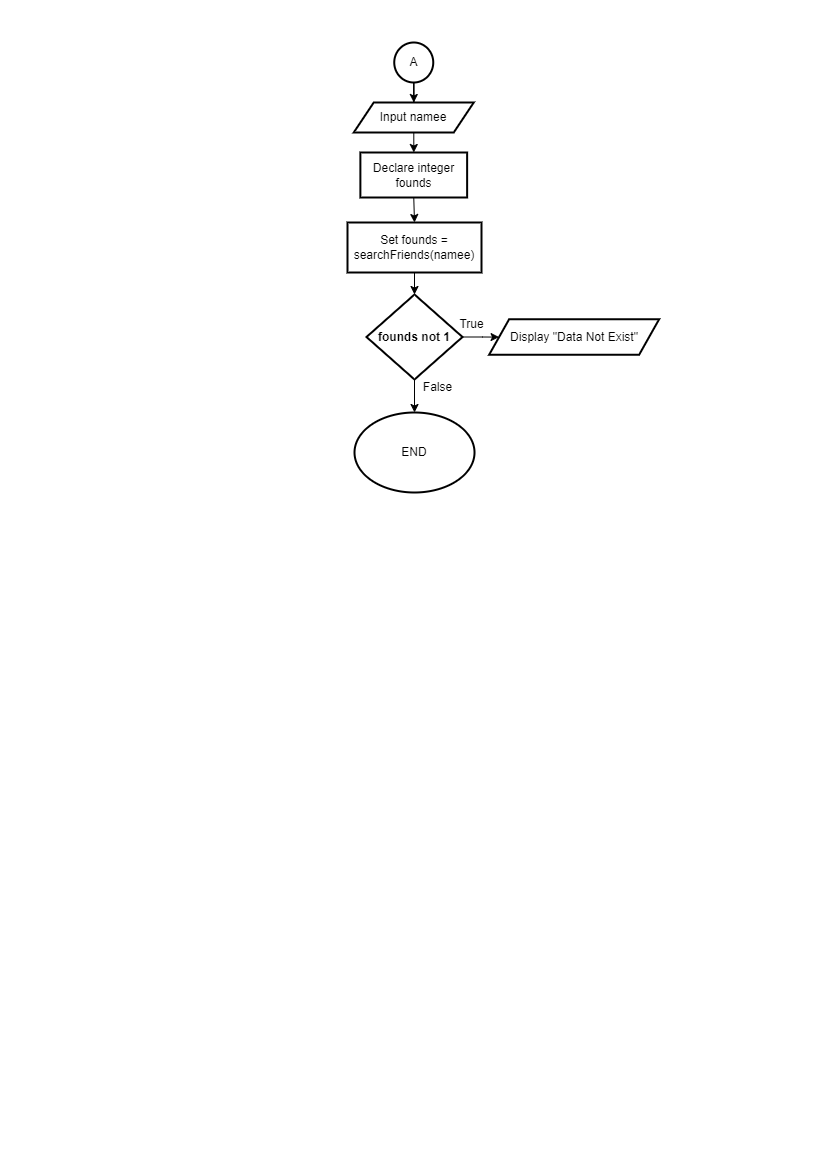
**End module**

**FLOWCHART**

**Diagram

Description automatically generated**

****

****

1. **PSEUDOCODE**

**Function** countArea(integer array matrix, integer n)

Declare integer x,y,area

Set area = 0;

**For x:=0 to n**

**For y:=0 to n**

**If matrix[x][y] == 1** **then**

Set area++

Set matrix[x][y] = 0

**If x > 0 AND matrix[x-1][y] == 1 then** set area+= countArea(matrix, n)

**End if**

**If x < n - 1 AND matrix[x+1][y] == 1** set area += countArea(matrix, n)

**End if**

**If y > 0 AND matrix[x][y-1] == 1 then** set area += countArea(matrix, n)

**End if**

**If y < n - 1 AND matrix[x][y+1] == 1 then** set area += countArea(matrix, n)

**End if**

**End if**

**End for**

**End for**

**Return** area

**End function**

**Module main()**

Declare integer N, i, j

Declare integer array mat1, mat2

Input N

**For i:=0 to N**

**For j:=0 to N**

Setmat1[i][j]=0

Set mat2[i][j]=0

**End for**

**End for**

**For i:=0 to N**

**For j:=0 to N**

Inputmat1[i][j]

**End for**

**End for**

**For i:=0 to N**

**For j:=0 to N**

Inputmat2[i][j]

**End for**

**End for**

Declare integer area1, area2

Setarea1 = countArea(mat1, N)

Setarea2 = countArea(mat2, N)

**If area1 > area2 then** Display “Object 1 is bigger”

**Else if area1 < area2 then** Display “Object 2 is bigger”

**Else** Display “Object 1 and 2 has the same size”

**End module**

**FLOWCHART**

**Diagram, engineering drawing

Description automatically generated**

Diagram

Description automatically generated

Diagram

Description automatically generated