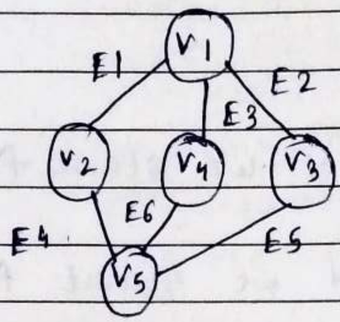
* Graph:-

- A graph is a collection of two sets V and E where V is a finite non-empty set of vertices and E is a finite non-empty set of edges.



$G = \{\{V_1, V_2, V_3, V_4, V_5\},$
$\{E1, E2, E3, E4, E5, E6\}\}$

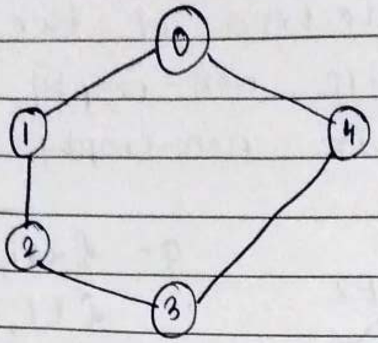
* BFS Traversal of Graph

- In BFS we start from some vertex and find all the adjacent vertices of it.
- This process will be repeated for all vertices so that the vertices lying on the same breadth get printed.
- For avoiding repetition of vertices, we maintain array of visited nodes.
- A queue data structure is used to store adjacent vertices.

* DFS Traversal of Graph

- In depth first search traversal we start from one vertex and traverse the path as deeply as we can go.
- When there is no vertex further, we traverse back and search for unvisited vertex.
- ~~The DFS~~ An array is maintained for storing the visited vertex.

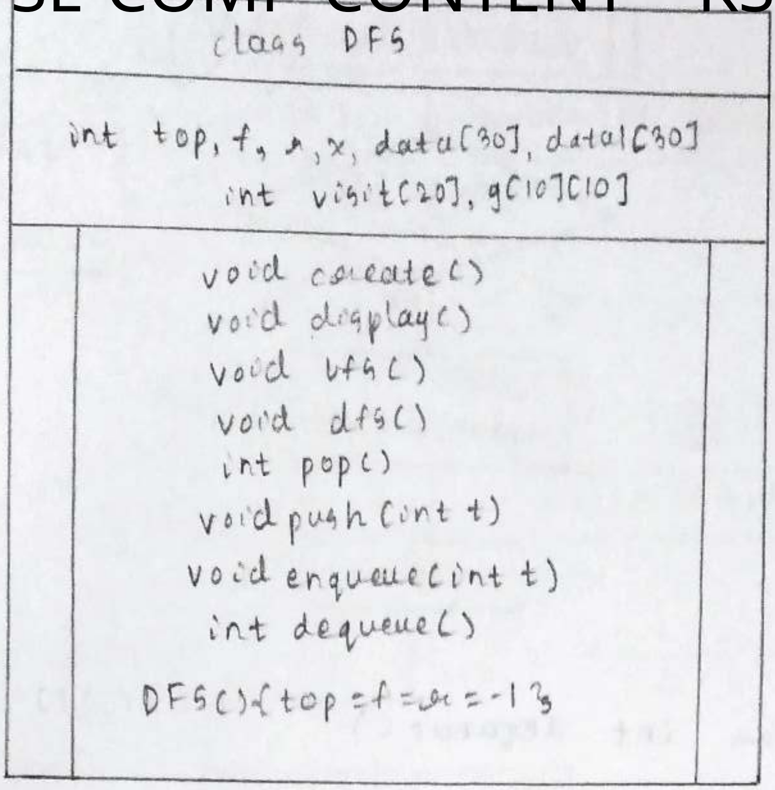
→ For ~~exam~~ example,



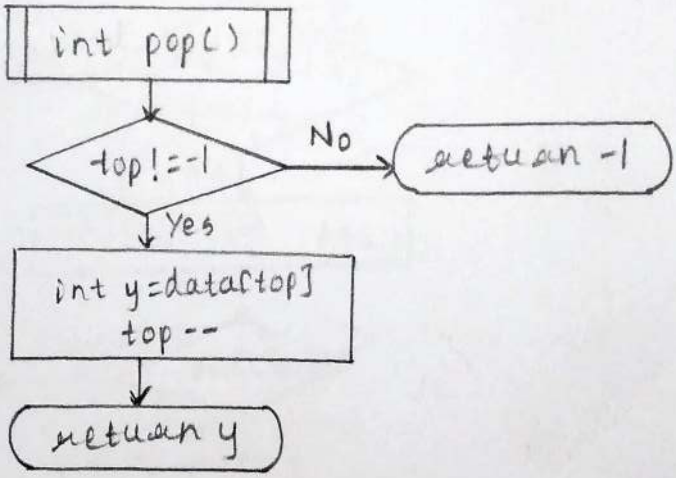
1) The DFS will be (if we start from vertex 0).
0-1-2-3-4

2) The DFS will be (if we start from vertex 3)
3-4-0-1-2

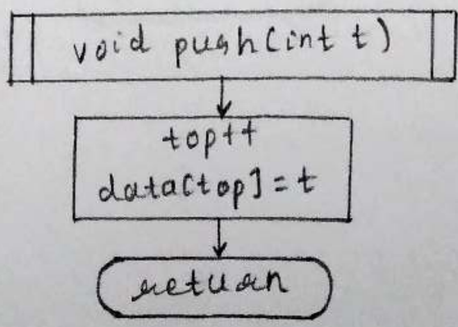
wchart for class DFS



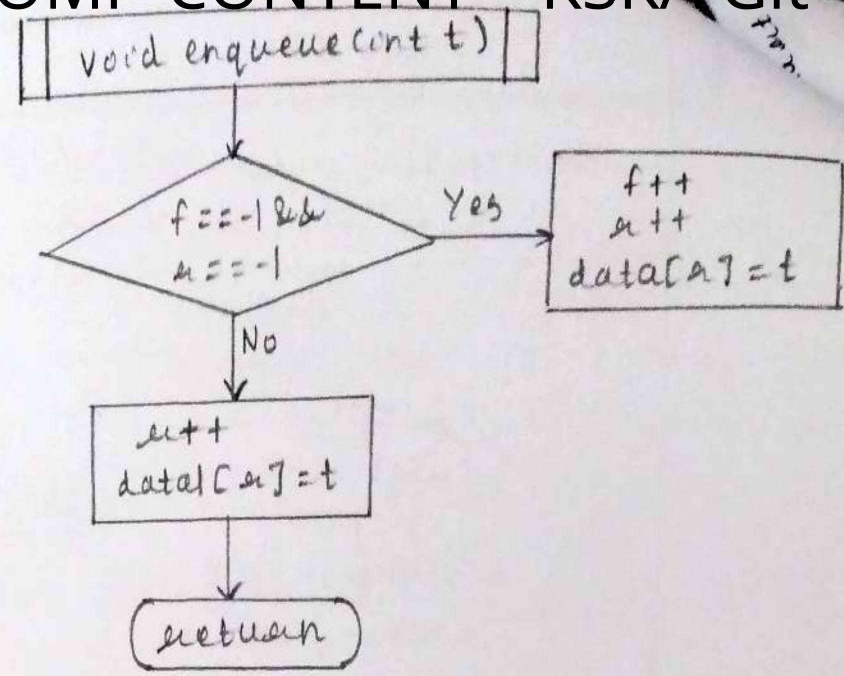
→ Flowchart for int pop()



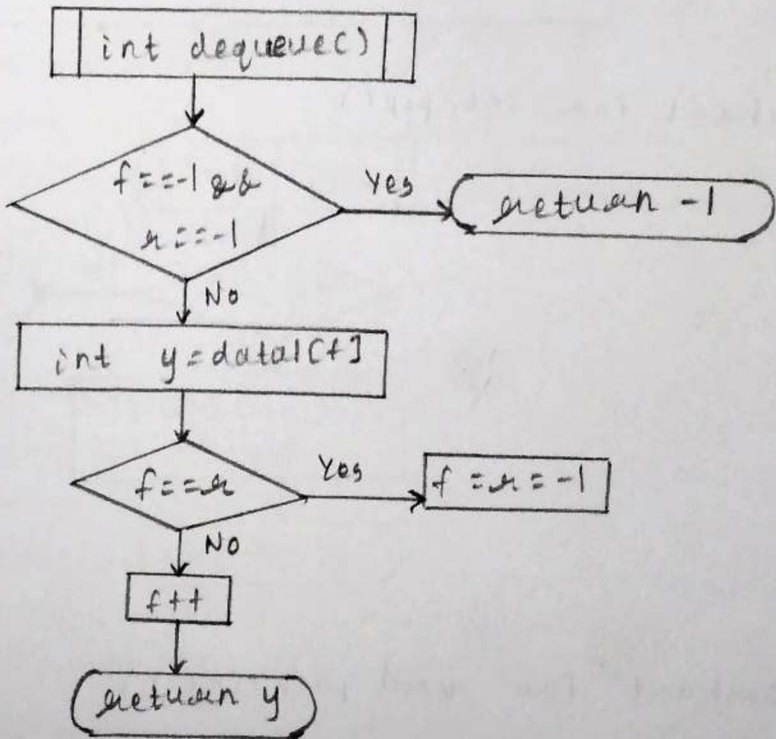
→ Flowchart for void push(int t)



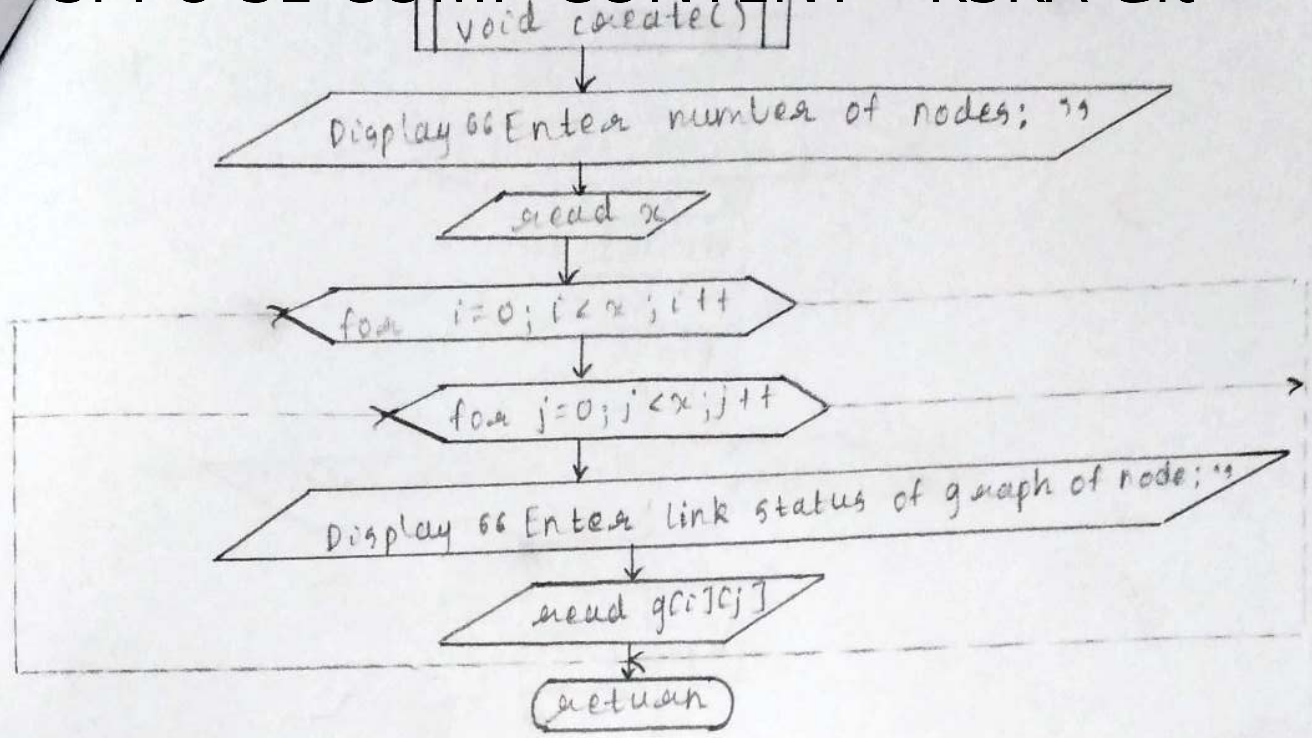
→ Flowchart for void enqueue (int t)



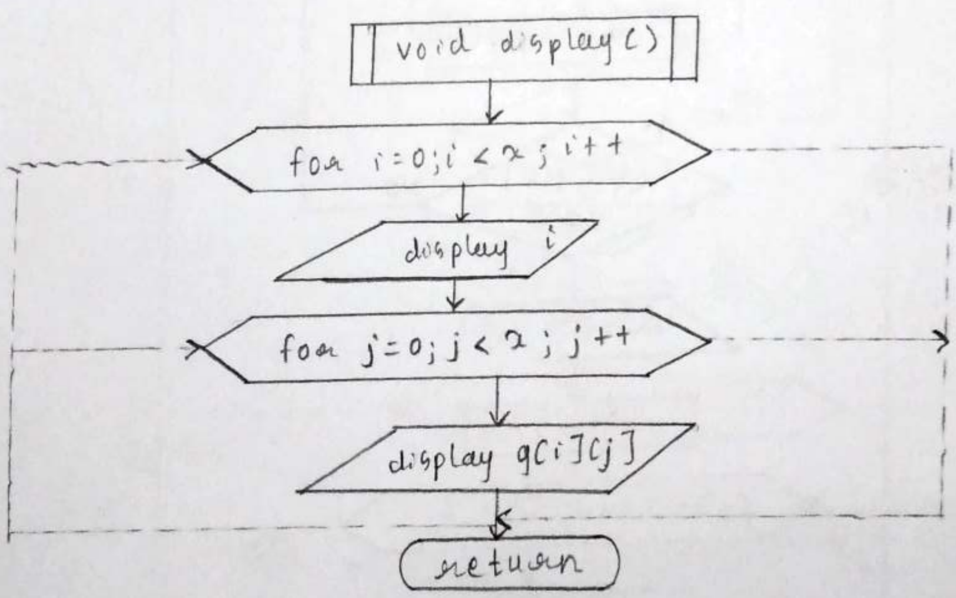
→ Flowchart for int dequeue()



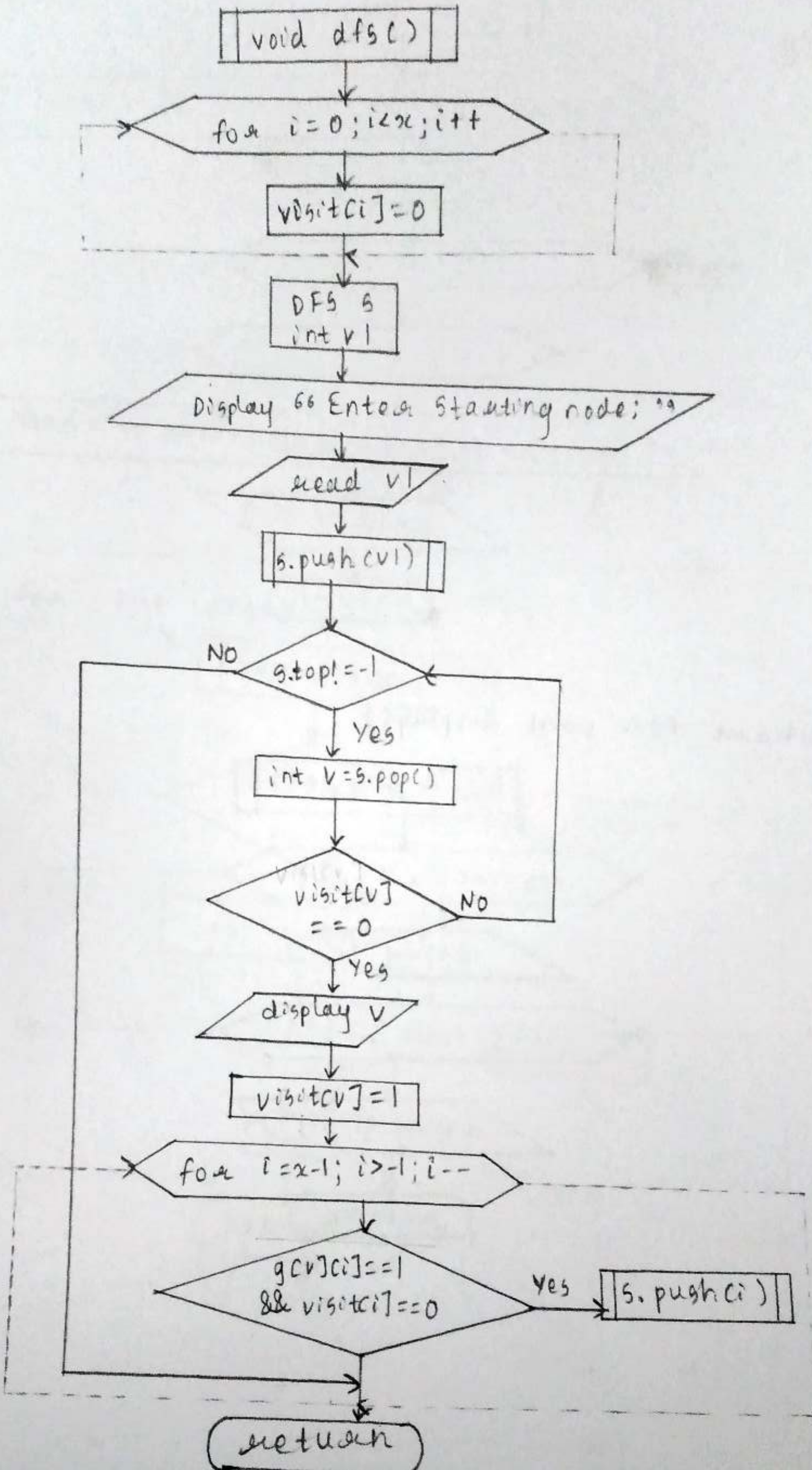
Flowchart for void create()



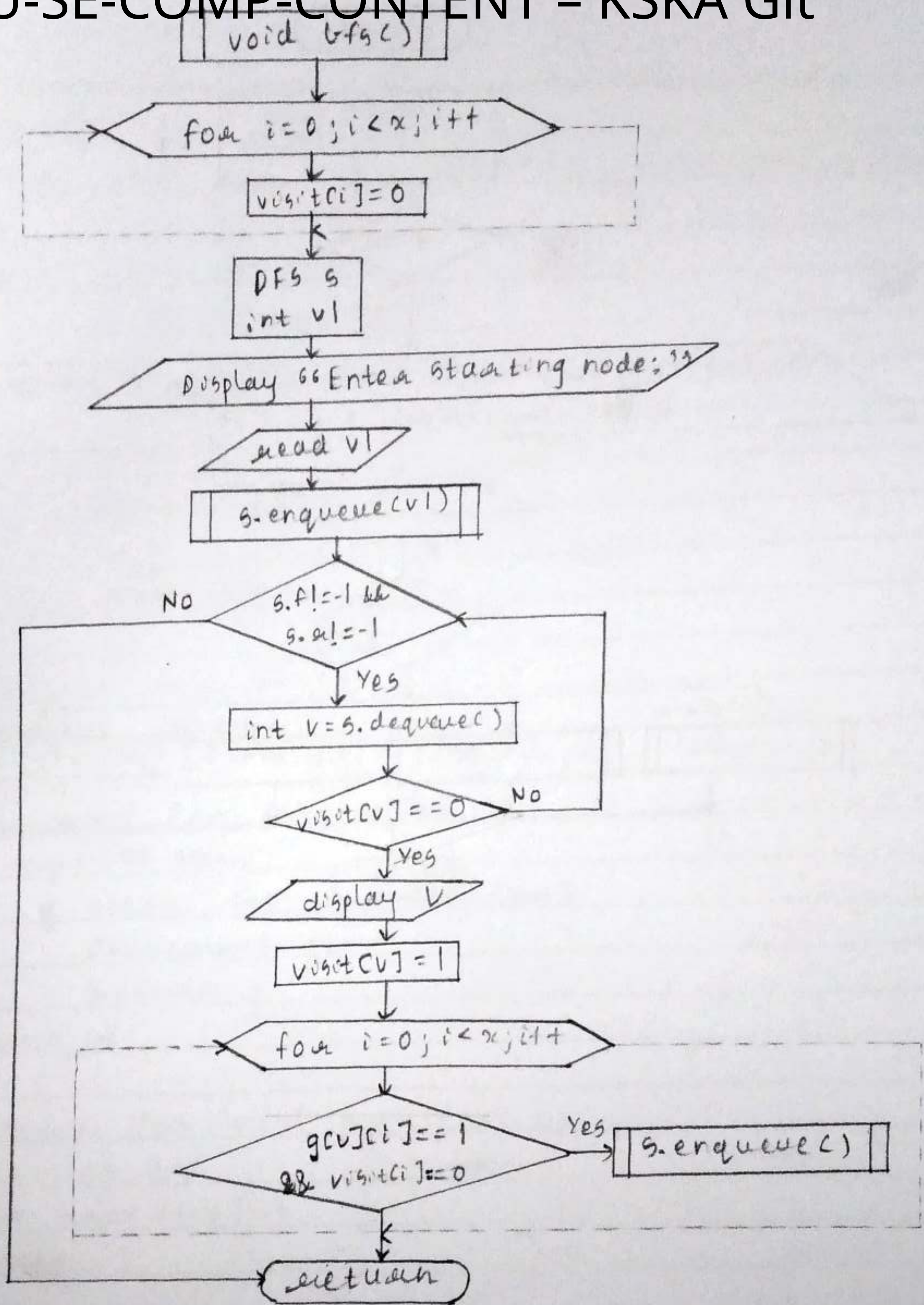
→ Flowchart for void display()



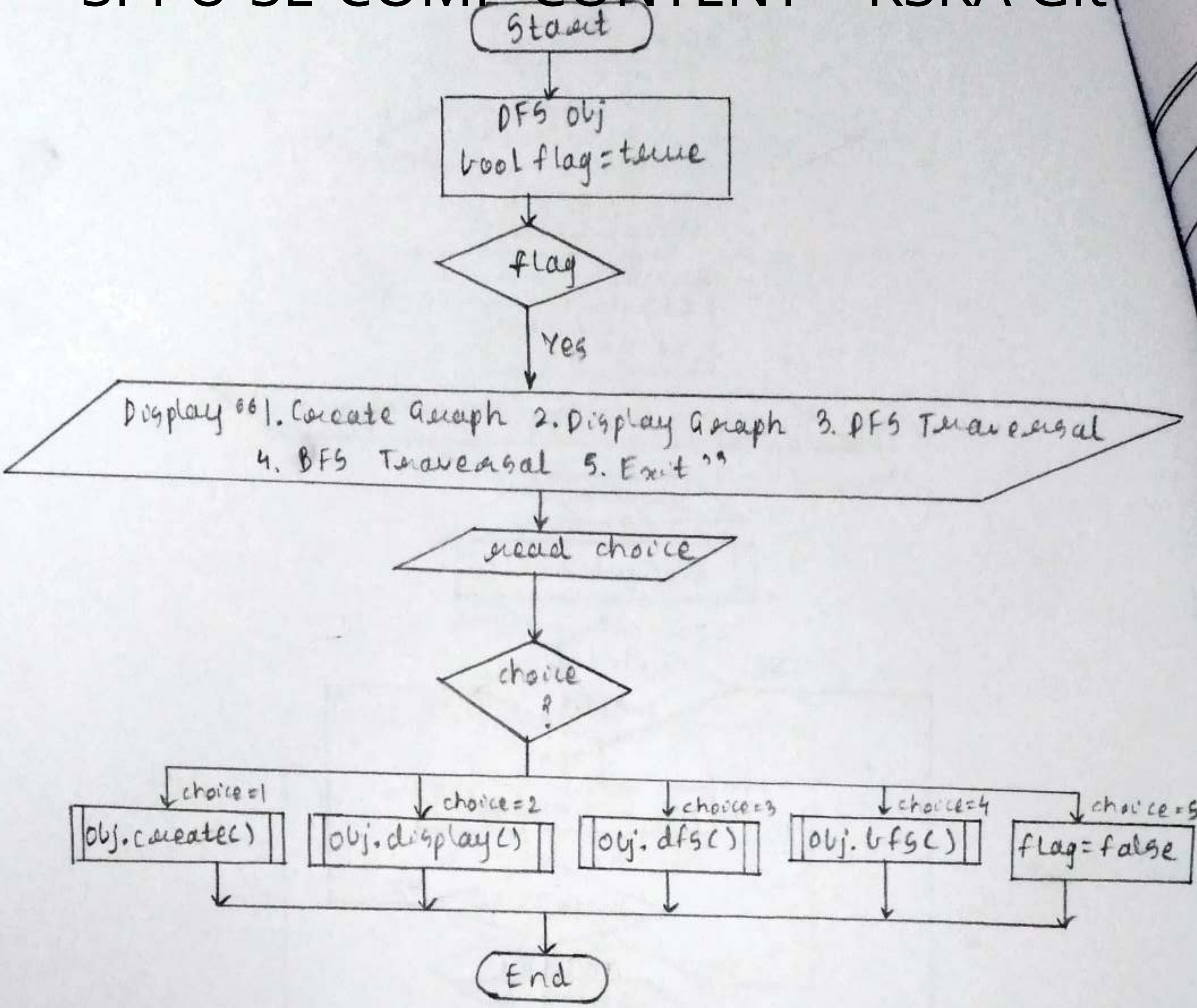
→ Flowchart for void dfs()



flowchart for void bfs()



→ Flowchart for int main()



→ Pseudocode for class DFS

1. Declare int top, f, r, x, data [30], data1[30], visit[20],
   int g[10][10]

2. Declare void create()
   void display()
   void dfs()
   void bfs()
   int pop()
   void push (int t)
   void enqueue (int t)
   ~~void~~ int dequeue ()

→ Pseudocode for DFS()

1. initialize top = f = r = -1

→ Pseudocode for int pop()

1. if top != -1 then
   store int y = data [top]
   decrement top
   return y
2. return -1

→ Pseudocode for void push (int t)

1. increment top
2. store data [top] = t
3. return

→ Pseudocode for void enqueue (int t)

1. if f == -1 and r == -1 then
   increment f and r
   store data1 [r] = t

else

increment r
store data1[r] = t
2. return

→ Pseudocode for int dequeue()
1. if f==-1 and r==-1 then
return -1
else
store int y=data1[f]
if f==r then
initialize f=r=-1
else
increment f
return y

→ Pseudocode for void create()
1. Read x
2. for i=0; i<x; i++ do
begin
for j=0; j<x; j++ do
begin
Display "Enter link status of graph of node: "
read g[i][j]
end
end
3. return

→ Pseudocode for void display()
1. for i=0; i<x; i++ do
begin
Display i

for j=0 ; j<x ; j++ do
begin
display g[i][j]
end
end
2. return

→ Pseudocode for void dfs()
1. for i=0 ; i<x; i++ do
begin
initialize visit[i]=0
end
2. Create DFS s
3. Declare int v1
4. read starting node v1
5. call function s.push(v1)
6. while s.top!=-1 do
begin
store int v=s.pop()
if visit[v]==0 then
display v
initialize visit[v]=1
for i=x-1; i>-1; i-- do
begin
if g[v][i]==1 and visit[i]==0 then
call function s.push(i)
end
end
7. return

→ Pseudocode for void bfs()
1. for i=0; i<x; i++ do
begin

initialize visit[i]=0

end

2. create DFS s

Declare int v1

3. read starting node v1

4. call function s.enqueue (v1)

5 while s.f!=-1 and s.r!=-1 do

begin

store int v=s.dequeue()

if visit[v]==0 then

display v

initialize visit[v]=1

for i=0; i<x; i++ do

begin

if g[v][i]==1 and visit[i]==0 then

call function s.enqueue()

end

end

6. return

→ Pseudocode for int main()

1. start

2. Create DFS obj

3. Declare bool flag=true

int choice

4. while flag do

begin

Display "1. Create Graph 2. Display Graph 3. DFS Traversal 4. BFS Traversal 5. Exit"

Display "Enter choice"

read choice

switch (choice)

case1:

call function obj.create()
break
case 2:
call function obj.display()
break
case 3:
call function obj.dfs()
break
case 4:
call function obj.bfs()
break
case 5:
store flag = false
break
default:
Display "Enter valid choice!"
break
end
5. End

Topic: ____________ Page No.: ____ Date.: / /

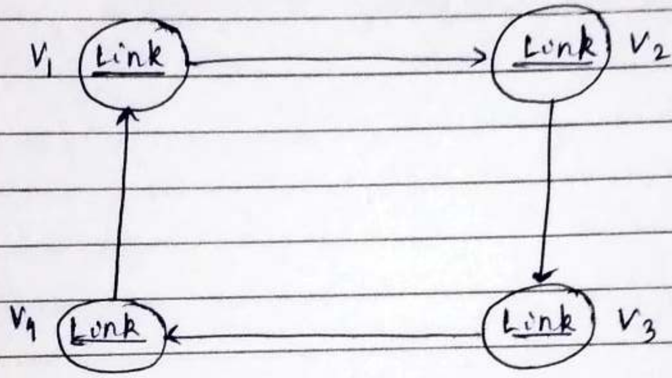
Q1. List applications of graph.

Ans.

1) Webgraph :-

- The webgraph is a directed graph, whose vertices are nothing but the web pages and the directed edges between any two vertices v_1 and v_2 exists if there is a hyperlink present on web page v_1 referring to page V_2

eg:



2) Page Rank :-

- It is an algorithm used for measuring the importance of website pages.

3) Google map :-

- Google map is a service developed by Google.
- It offers services for satellite imagery, street maps, 360° views of streets and real-time conditions.

4) Network monitoring :-

- Graphs can be used to monitor network traffic in real time, allowing network administrators to identify potential bottlenecks.

5) Biology :-

- Graphs are used to model biochemical reactions, genetic interactions, and neural networks.

Q2. Given an undirected graph G with V vertices and E edges, what is the sum of the degrees of all vertices.

Ans. :- Consider an undirected graph G with V vertices and E edges.

- Let the degree of vertex i be d_i.
- This is because each edge contributes two to the sum of degrees, one for each of its endpoints.
- Therefore, the sum of degrees of all vertices in an undirected graph with V vertices and E edges is 2E.