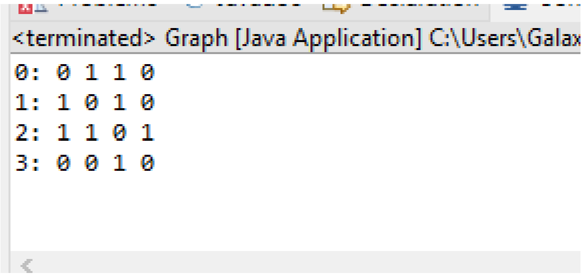
Q1. Implement a graph using adjacency matrix.

```
package com.graph;
public class Graph {
private boolean adjMatrix[][];
private int numVertices;
public Graph(int numVertices) {
this.numVertices = numVertices;
adjMatrix = new boolean[numVertices][numVertices];
}
public void addEdge(int i, int j) {
adjMatrix[i][j] = true;
adjMatrix[j][i] = true;
}
public void removeEdge(int i, int j) {
adjMatrix[i][j] = false;
adjMatrix[j][i] = false;
}
public String toString() {
StringBuilder s = new StringBuilder();
for (int i = 0; i < numVertices; i++) {
s.append(i + ": ");
for (boolean j : adjMatrix[i]) {
s.append((j ? 1 : 0) + " ");
}
s.append("\n");
}

return s.toString();

}

public static void main(String args[]) {

Graph g = new Graph(4);

g.addEdge(0, 1);

g.addEdge(0, 2);

g.addEdge(1, 2);

g.addEdge(2, 0);

g.addEdge(2, 3);

System.out.print(g.toString());

}

}
```

Output:



```
<terminated> Graph [Java Application] C:\Users\Galax
0: 0 1 1 0
1: 1 0 1 0
2: 1 1 0 1
3: 0 0 1 0
```

Q2. Implement stack using linked list

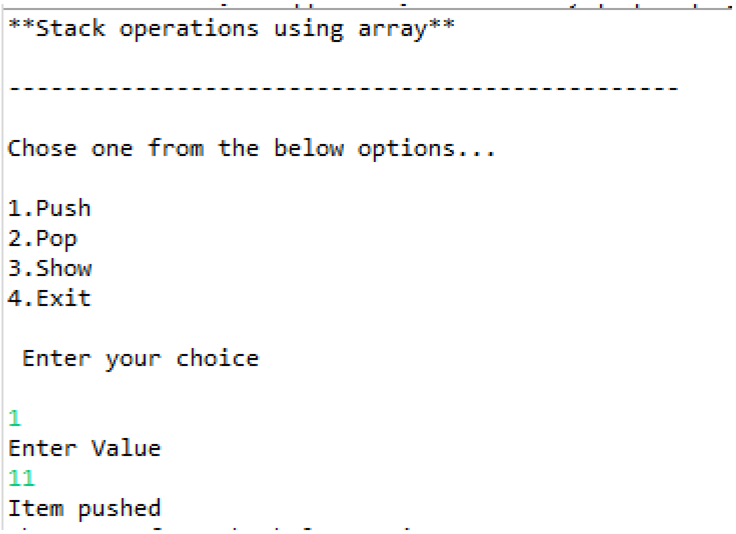
```
package com.stack;

import java.util.Scanner;

public class Stack {

int top;

int maxsize = 10;
```

```java
}
int[] arr = new int[maxsize];
boolean isEmpty()
{
return (top < 0);
}
Stack()
{
top = -1;
}
boolean push (Scanner sc)
{
if(top == maxsize-1)
{
System.out.println("Overflow !!");
return false;
}
else
{
System.out.println("Enter Value");
int val = sc.nextInt();
top++;
arr[top]=val;
System.out.println("Item pushed");
return true;
}
```

```
}
boolean pop ()
{
if (top == -1)
{
System.out.println("Underflow !!");
return false;
}
else
{
top --;
System.out.println("Item popped");
return true;
}
}
void display ()
{
System.out.println("Printing stack elements .....");
for(int i = top; i>=0;i--)
{
System.out.println(arr[i]);
}
}
public static void main(String[] args) {
int choice=0;
```

```
Scanner sc = new Scanner(System.in);

Stack s = new Stack();

System.out.println("**Stack operations using array**\n");

System.out.println("------------------------------------------------\n");

while(choice != 4)

{

System.out.println("Chose one from the below options...\n");

System.out.println("1.Push\n2.Pop\n3.Show\n4.Exit");

System.out.println("\n Enter your choice \n");

choice = sc.nextInt();

switch(choice)

{

case 1:

{

s.push(sc);

break;

}

case 2:

{

s.pop();

break;

}

case 3:

{

s.display();
break;

}

case 4:

{

System.out.println("Exiting....");

System.exit(0);

break;

}

default:

{

System.out.println("Please Enter valid choice ");

}

}

}

}

}
```

Output:



```
**Stack operations using array**

------------------------------------------------

Chose one from the below options...

1.Push
2.Pop
3.Show
4.Exit

 Enter your choice

1
Enter Value
11
Item pushed
```