Java Programming

-10 December 2019
-Aadhitya Swarnesh I
-[redacted]

Question 1 :

```
class q1
{
    boolean isPrime(int a)
    {
        int i;
        for(i=2;i<(int)(a/2);i++)
        {
            if(a%i==0)
            {
                return(false);
            }
        }
        return(true);
    }

    int findSum(int a)
    {
        int b, s=0;
        while(a>0)
        {
            b = a%10;
            a = a/10;
            s = s+b;
        }
        return(s);
    }
    public static void main(String[] args)
    {
        int i,s, c=0;
        q1 ob = new q1();
        for(i=1000; i<2000;i++)
        {
            s = ob.findSum(i);
            if(ob.isPrime(s))
            {
                c++;
            }
        }
        System.out.println("The number of primes are : " + c);
    }
}
```

```
(base) Aadhityas-MacBook-Air:JP aadhitya$ javac q1.java
(base) Aadhityas-MacBook-Air:JP aadhitya$ java q1
The number of primes are : 344
(base) Aadhityas-MacBook-Air:JP aadhitya$
```

Question 2 :

```
import java.io.*;
class q2
{
    int adj[][];
    void printHamiltonian(int visited[], int numVisited, int totNodes, int curNode, int path[], int pathl)
    {
        int i;
        visited[curNode] = 1;
        numVisited++;
        if(numVisited==totNodes)
        {
            for(i=0;i<pathl;i++)
            {
                System.out.print((path[i]+1));
            }
            System.out.println();
        }
        else
        {
            for(i=0;i<totNodes;i++)
            {
                if(visited[i]==0 && adj[curNode][i]==1)
                {
                    path[pathl] = i;
                    printHamiltonian(visited, numVisited, totNodes, i, path, pathl+1);
                }
            }
        }
        visited[curNode] = 0;
    }

    public static void main(String args[]) throws IOException
    {
        q2 ob = new q2();
        InputStreamReader read = new InputStreamReader(System.in);
        BufferedReader in = new BufferedReader(read);
        int i, n, j, e;
        System.out.print("Enter the number of nodes : ");
        n = Integer.parseInt(in.readLine());
        ob.adj = new int[n][n];
        int a[] = new int[2];
        String ss[] = new String[2];

        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                ob.adj[i][j] = 0;
            }
        }

        System.out.print("Enter the number of edges : ");
        e = Integer.parseInt(in.readLine());
        for(i=0;i<e;i++)
        {
            System.out.println("Enter the source and destination node for edge " + (i+1));
```

```
            ss = in.readLine().split(" ");
            a[0] = Integer.parseInt(ss[0]) - 1;
            a[1] = Integer.parseInt(ss[1]) - 1;
            ob.adj[a[0]][a[1]] = 1;
            ob.adj[a[1]][a[0]] = 1;
        }

        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                System.out.print(ob.adj[i][j]);
            }
            System.out.println();
        }

        int v[] = new int[n];
        for(i=0;i<n;i++)
        {
            v[i] = 0;
        }
        int path[] = new int[n];
        System.out.println("Hamiltonian Paths :");
        for(i=0;i<n;i++)
        {
            path[0] = i;
            ob.printHamiltonian(v, 0, n, i, path, 1);
        }

    }
}
```

```
(base) Aadhityas-MacBook-Air:JP aadhitya$ javac q2.java
(base) Aadhityas-MacBook-Air:JP aadhitya$ java q2
Enter the number of nodes : 4
Enter the number of edges : 3
Enter the source and destination node for edge 1
1 2
Enter the source and destination node for edge 2
2 3
Enter the source and destination node for edge 3
1 4
0101
1010
0100
1000
Hamiltonian Paths :
3214
4123
(base) Aadhityas-MacBook-Air:JP aadhitya$
```