Name: Chavda Vivek M.

Enrol no: 92200133026

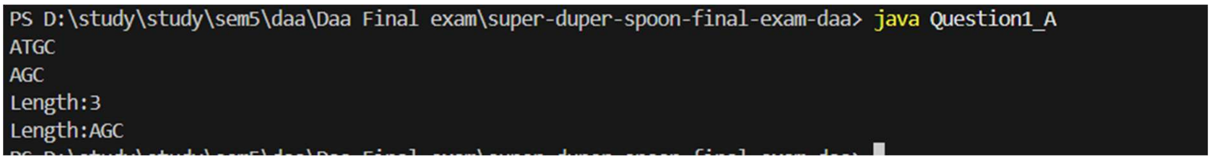
Question 1 a

Code

```
import java.util.Scanner;


public class Question1_A {

    public static int LCS(String X ,String Y)

    {
        int m = X.length();
        int n = Y.length();

        int[][] dp = new int[m+1][n+1];
        for(int i = 1;i<=m;i++){
            for(int j = 1;j<=n;j++){
                if(X.charAt(i-1)==Y.charAt(j-1)){
                    dp[i][j]=dp[i-1][j-1]+1;

                }else{
                    dp[i][j] = Math.max(dp[i-1][j],dp[i][j-1]);
                }
            }
        }
        return dp[m][n];
    }
    public static String LCSeq(String X ,String Y)
```

```
{
    int m = X.length();
    int n = Y.length();

    int[][] dp = new int[m+1][n+1];
    for(int i = 1;i<=m;i++){
        for(int j = 1;j<=n;j++){
            if(X.charAt(i-1)==Y.charAt(j-1)){
                dp[i][j]=dp[i-1][j-1]+1;

            }else{
                dp[i][j] = Math.max(dp[i-1][j],dp[i][j-1]);
            }
        }
    }
    StringBuilder LCS = new StringBuilder();
    int i = m , j = n;
    while(i>0 && j>0){
        if(X.charAt(i-1)==Y.charAt(j-1)){
            LCS.insert(0, X.charAt(i-1));
            i--;
            j--;
        }else if(dp[i-1][j]>dp[i][j-1]){
            i--;
        }else{
            j--;
        }
    }
    return LCS.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String x,y;
        x = sc.nextLine();
        y = sc.nextLine();
        int lcs =  LCS(x,y);
        String lcs1 = LCSeq(x,y );
        System.out.println("Length:"+lcs);
        System.out.println("Length:"+lcs1);
    }

}
```

Output:



```
PS D:\study\study\sem5\daa\Daa Final exam\super-duper-spoon-final-exam-daa> java Question1_A
ATGC
AGC
Length:3
Length:AGC
```

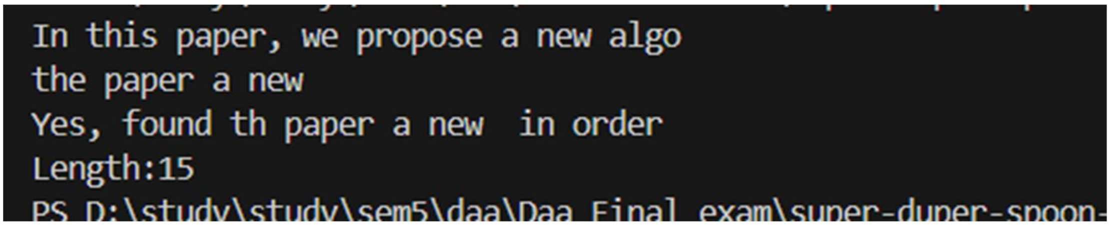
Question 1 b

Code:

```
import java.util.Scanner;

public class Question2_A {
    public static int LCS1(String X ,String Y)

    {
        int m = X.length();
        int n = Y.length();
    int[][] dp = new int[m+1][n+1];
    for(int i = 1;i<=m;i++){
        for(int j = 1;j<=n;j++){
            if(X.charAt(i-1)==Y.charAt(j-1)){
                dp[i][j]=dp[i-1][j-1]+1;

            }else{
                dp[i][j] = Math.max(dp[i-1][j],dp[i][j-1]);
            }
        }
    }
    return dp[m][n];
}
public static void LCSeq1(String X ,String Y)

{
    int m = X.length();
    int n = Y.length();

    int[][] dp = new int[m+1][n+1];
    for(int i = 1;i<=m;i++){
        for(int j = 1;j<=n;j++){
            if(X.charAt(i-1)==Y.charAt(j-1)){
                dp[i][j]=dp[i-1][j-1]+1;

            }else{
                dp[i][j] = Math.max(dp[i-1][j],dp[i][j-1]);
            }
        }
    }
    StringBuilder LCS = new StringBuilder();
```

```
        int i = m , j = n;
        while(i>0 && j>0){
            if(X.charAt(i-1)==Y.charAt(j-1)){
                LCS.insert(0, X.charAt(i-1));
                i--;
                j--;
            }else if(dp[i-1][j]>dp[i][j-1]){
                i--;
            }else{
                j--;
            }
        }


        System.out.println("Yes, found " +LCS.toString()+" in order");



    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String x,y;
        x = sc.nextLine();
        y = sc.nextLine();
        int lcs =  LCS1(x, y);
        LCSeq1(x, y);
        System.out.println("Length:"+lcs);

    }
}
```

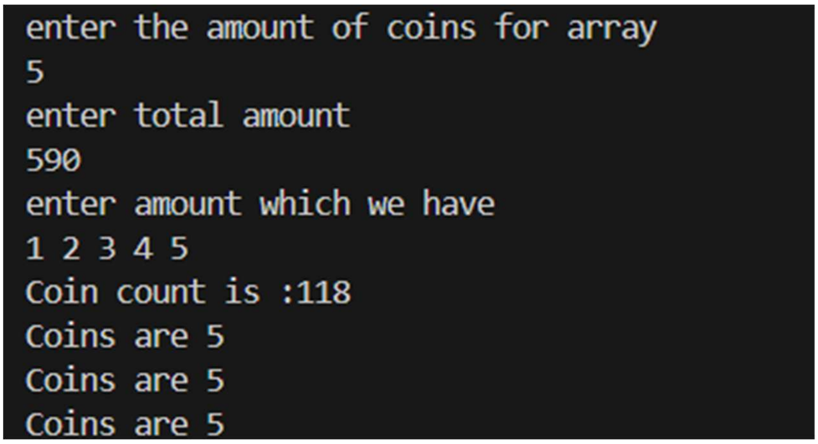
Output:



```
In this paper, we propose a new algo
the paper a new
Yes, found th paper a new  in order
Length:15
PS D:\study\study\sem5\daa\Daa Final exam\super-duper-spoon-
```

Question 2 A

```
import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;


public class Question1_B {

  public static void Coin(int[] coins ,int amount){

     Arrays.sort(coins);

     ArrayList<Integer> ans = new ArrayList<Integer>();

     int coinCount = 0;


     for(int i = coins.length-1 ; i >=0 ; i--){

        if(coins[i] <= amount){

        while(coins[i]<=amount){

           coinCount++;

           ans.add(coins[i]);

           amount -= coins[i];

      }

    }

    }

    System.out.println("Coin count is :" + coinCount);


    for(int i = 0 ; i < ans.size() ; i++){

       System.out.println("Coins are "+ans.get(i));

    }
  }
   public static void main(String[] args) {
     Scanner sc  = new Scanner(System.in);
     System.out.println("enter the amount of coins for array");
      int n = sc.nextInt();
      System.out.println("enter total amount");
      int amount = sc.nextInt();;
      int[] coins =new int[n];
      System.out.println("enter amount which we have ");
      for(int i = 0 ; i < coins.length ; i++){
          coins[i] = sc.nextInt();
      }
      Coin(coins, amount);

  }
}
```

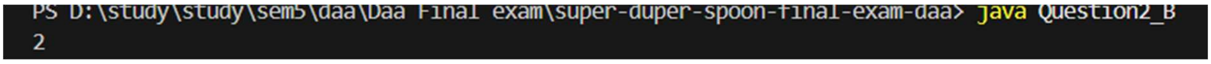


```
enter the amount of coins for array
5
enter total amount
590
enter amount which we have
1 2 3 4 5
Coin count is :118
Coins are 5
Coins are 5
Coins are 5
```

Question2 b

```
// Source code is decompiled from a .class file using FernFlower decompiler.
public class Question2_B {
  public Question2_B() {
  }

  public static int findMaxCount(int var0, int var1, int var2, int var3) {
    int var4 = 0;
    if (var0 % var1 != 0) {
      var0 /= var1;
      ++var4;
    }

    if (var0 % var2 != 0) {
      var0 /= var2;
      ++var4;
    }

    if (var0 % var3 != 0) {
      int var10000 = var0 / var3;
      ++var4;
    }

    return var4 != 0 ? var4 : 0;
  }

  public static void main(String[] var0) {
    System.out.println(findMaxCount(5, 5, 3, 2));
  }
}
```

Output:



```
PS D:\study\study\sem5\daa\Daa Final exam\super-duper-spoon-final-exam-daa> java Question2_B
2
```

Question 3 A

Code:

```
import java.util.*;

public class Question3_A {

    static class Edge implements Comparable<Edge> {
        int src;
        int dest;
        int wt;

        public Edge(int s, int d, int w) {
            this.src = s;
            this.dest = d;
            this.wt = w;
        }

        @Override
        public int compareTo(Edge e2) {
            return this.wt - e2.wt;
        }

    }

    static void createGraph(ArrayList<Edge> edges) {

        edges.add(new Edge(0, 1, 10));
        edges.add(new Edge(0, 2, 15));
        edges.add(new Edge(0, 3, 30));
        edges.add(new Edge(1, 3, 40));
        edges.add(new Edge(2, 3, 50));
```

```
}

static int n = 4;
static int par[] = new int[n];
static int rank[] = new int[n];

public static void initialise() {
    for (int i = 0; i < n; i++) {
        par[i] = i;
    }
}

public static int find(int x) {
    if (par[x] == x) {
        return x;
    }
    return par[x] = find(par[x]);
}

public static void union(int a, int b) {
    int parA = find(a);
    int parB = find(b);

    if (rank[parA] == rank[parB]) {
        par[parB] = parA;
        rank[parA]++;

    } else if (rank[parA] < rank[parB]) {
        par[parA] = parB;
    } else {
        par[parB] = parA;
    }
}

public static void Kruskals(ArrayList<Edge> edges, int V) {
    initialise();

    Collections.sort(edges);
    int mstCost = 0;
    int count = 0;

    for (int i = 0; i < V - 1; i++) {
        Edge e = edges.get(i);

        int parA = find(e.src);
        int parB = find(e.dest);

        if (parA != parB) {
            union(e.src, e.dest);
            mstCost += e.wt;
        }
    }

    System.out.println(mstCost);
}

public static void main(String args[]) {
    int V = 4;
    ArrayList<Edge> edges = new ArrayList<>();
    createGraph(edges);
```

```
        Kruskals(edges,V);

    }
}
```

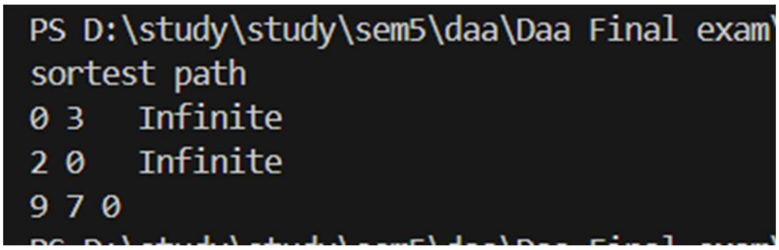
Question 3 b

Code:

```
import java.util.Scanner;

public class Question3_B {
    static  final int INF = 99999;
    public static void floydWarshall(int[][] graph){
        int V= graph.length;
        int[][] dist = new int[V][V];

        for(int i=0;i<V;i++){
            for(int j=0;j<V;j++){
                dist[i][j] = graph[i][j];

            }

        }
        for(int k = 0 ; k<V;k++){
            for(int i = 0 ; i<V;i++){
             for(int j = 0 ; j<V;j++){
                if(dist[i][k]+dist[k][j] < dist[i][j]){
                    dist[i][j] = dist[i][k]+dist[k][j];
                }
             }
        }       }
        printlnSolution(dist);


}
public static void printlnSolution(int[][] dist){
    int V = dist.length;
    System.out.println("sortest path");
    for(int i = 0; i<V;i++){
        for(int j = 0;j<V;j++){
            if(dist[i][j]==INF){
                System.out.print("  Infinite");
            }else{
                System.out.print(dist[i][j]+ " ");
            }
        }
        System.out.println();
    }
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    // System.out.println("n size");
    // int n = sc.nextInt();
    // System.out.println("m size");
    // int m = sc.nextInt();


    // int[][] graph = new int[n][m];
    // for(int i = 0 ; i<n ; i++){
    //    for(int j = 0 ; j<n ; j++){
    //        graph[i][j ] =sc.nextInt();
    //    }
    // }
```

```
        int graph[][] = {

            {0,3,INF,INF,INF,INF},
            {2,0,INF,INF,INF,INF},
            {INF,7,0,1,INF,2},



            };

        floydWarshall(graph);
    }
}
```

Output:



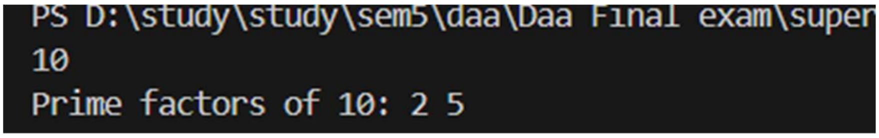
```
PS D:\study\study\sem5\daa\Daa Final exam
sortest path
0 3   Infinite
2 0   Infinite
9 7 0
```

Base own my input

Question 4

Code

```
import java.util.Scanner;

public class Question4 {
    public static void main(String[] args) {
        int num;
        Scanner sc = new Scanner(System.in);
        num = sc.nextInt();
        System.out.println("Prime factors of " + num + ": " + primeFactors(num));
    }

    public static String primeFactors(int n) {
        StringBuilder sb = new StringBuilder();
        for (int i = 2; i <= n; i++) {
            while (n % i == 0) {
                sb.append(i).append(" ");
                n /= i;
            }
        }
        return sb.toString().trim();
    }
}
```



```
PS D:\study\study\sem5\daa\Daa Final exam\super
10
Prime factors of 10: 2 5
```