



eduresourceza.com

G11 PRAC MEMO NOV 2019

SQL

1.	SELECT DISTINCT✓ Commander FROM tblMission✓ WHERE LunarLanding = true✓ ORDER BY Commander; ✓	(4)
2.	SELECT MissionName, Distance✓ FROM tblMission ORDER BY Distance ✓DESC ✓ LIMIT 3 ✓	(4)
3.	SELECT LaunchSite, COUNT(*)✓ AS NumMissions✓ FROM tblMission GROUP BY✓ LaunchSite✓ ORDER BY NumMissions ✓DESC; ✓	(6)
4.	SELECT YEAR(LaunchDate) ✓AS YEAR, COUNT(*)✓ AS NumMissions✓ FROM tblMission GROUP BY✓ YEAR(LaunchDate), ✓	(5)
5.	SELECT ROUND(✓EarthRevolutions + LunarRevolutions✓)/2,2) ✓ AS TOTALRevolutions FROM tblMission WHERE MissionType✓ LIKE "CREWED*";✓	(5)
6.	SELECT Commander, ✓ CONCAT✓ (LEFT✓ (Commander,3), ✓ FLOOR(RAND()*✓ (600-500+1) + 500)) ✓ AS Code FROM tblMission;	(6)
7.	UPDATE tblMission✓ SET MissionName = "Apollo 11" ✓ Where MissionName = 'Apolo 11'; ✓	(3)
8.	SELECT Commander, SUM✓ (Distance) ✓ AS TotalDist✓ FROM tblMission GROUP BY✓ Commander✓ HAVING✓ TotalDist > 1000000; ✓	(7)

OOP MEMO

//2.1 [3]

```
public class Astronaut✓
{
```

```
    private int missionNo;
    private String dates;
    private boolean landing;
    private String fullName; ✓✓
```

//2.2 [2]

```
    public Astronaut(int m, String d, boolean l, String n) ✓
    {
        missionNo = m;
        dates = d;
        landing = l;
        fullName = n; ✓
    }
```

//2.3 [1]

```
    public int getMissionNo()
    {
        return missionNo; ✓
    }
```

//2.4 [5]

```
    public String reorderName()✓
    {
        String tempArr[] = fullName.split(" ");✓
        return tempArr[1] ✓ + ", "✓ + tempArr[0]; ✓
    }
```

//2.5 [4]

```
    @Override
    public String toString()
    {
        String temp = "";
        if (landing) ✓
        {
            temp = "Lunar Mission ";✓
        }
        return temp + "Apollo " + missionNo ✓+ "\t\t" + dates
+ "\t" + reorderName();✓
    }
}
```

//3.1 [2]

```
public class Mission✓
{
    private String name;
    private String crew; ✓
```

//3.2 [2]

```
    public Mission(String n, String c) ✓
    {
        name = n;
        crew = c; ✓
    }
```

//3.3 [1]

```
    public String getCrew()
    {
        return crew; ✓
    }
```

//3.4 [3]

```
    @Override
    public String toString() ✓
    {
        return "Mission: " + name ✓ + "\tCrew: " + crew; ✓
    }
}
```

//4.1 [1]

```
public class MissionArray✓
{
```

//4.2 [3]

```
    private Mission[] mArr ✓ = new Mission[20]; ✓
    private int mSize = 0; ✓
```

//4.3 [1]

```
    public MissionArray() ✓
    {
    }
}
```

//4.4 [4]

```
    public void addMission(String n, String c) ✓
    {
        mArr[mSize] ✓ = new Mission(n, c); ✓
        mSize++; ✓
    }
```

//8.1 [6]

```

public int countMissions(String n) ✓
{
    int count = 0;
    for (int i = 0; i < mSize; i++) ✓
    {
        if (mArr[i].getCrew().contains(n)) ✓✓
        {
            count++; ✓
        }
    }
    return count; ✓
}

```

//4.5 [5]

```

public String toString() ✓
{
    String rString = "";
    for (int i = 0; i < mSize; i++) ✓
    {
        rString += mArr[i].toString() + "\n"; ✓
    }

    return rString; ✓
}

```

//5.1 [1]

```

public class AstroArray ✓
{

```

//5.2 [3]

```

    private Astronaut[] astroArr = new Astronaut[50]; ✓
    private int size = 0; ✓
    private MissionArray ma = new MissionArray(); ✓

```

//5.3 [7]

```

public AstroArray()
{
    try
    {
        Scanner scFile = new Scanner(new
        FileReader("Astronauts.txt")); ✓
        String d, n;
        int m;
        boolean l;
        int t;
        while (scFile.hasNextLine()) ✓
        {
            String line = scFile.nextLine(); ✓

```

```

        Scanner scLine = new
Scanner(line).useDelimiter(","); ✓

        m = scLine.nextInt();
        d = scLine.next();
        l = scLine.nextBoolean();
        n = scLine.next(); ✓

        astroArr[size] = new Astronaut(m, d, l, n); ✓
        size++; ✓
    }
    scFile.close();
} catch (FileNotFoundException ex)
{
    System.out.println("File not found");
}

}
//5.4 [3]
public String toString()
{
    String rString = "";
    for (int i = 0; i < size; i++) ✓
    {
        rString += astroArr[i].toString() + "\n"; ✓
    }

    return rString; ✓
}
//5.5 [6]
public void sortAstronaut()
{
    for (int i = 0; i < size - 1; i++) ✓
    {
        for (int j = i + 1; j < size; j++) ✓
        {
            if
(astroArr[i].reorderName() ✓.compareTo(astroArr[j].reorderName(
)) > 0) ✓
            {
                Astronaut temp = astroArr[i];
                astroArr[i] = astroArr[j];
                astroArr[j] = temp; ✓✓
            }
        }
    }
}
}
//7.1 [8]

```

```

public MissionArray✓ createMissionArray()
{
    String name = "";
    String crew = "";
    for (int i = 7; i <= 17; i++)✓
    {
        crew = "";
        for (int j = 0; j < size; j++)✓
        {
            name = "Apollo " + i; ✓
            if (astroArr[j].getMissionNo() == i) ✓
            {
                crew = crew + astroArr[j].reorderName() +
"; "✓
            }
        }
        ma.addMission(name, crew); ✓
    }
    return ma; ✓
}

```

//6.1 [1]

```

public class AstroUI✓
{

```

```

    public static void main(String[] args)
    {

```

//6.2 [1]

```

        AstroArray aArr = new AstroArray();✓

```

//6.3 [2]

```

        aArr.sortAstronaut();✓
        System.out.println("Astronauts sorted by surname\n" +
aArr); ✓

```

//7.2 [3]

```

        MissionArray ma✓ = aArr.createMissionArray();✓
        System.out.println(ma); ✓

```

//8.2 [2]

```

        String name = JOptionPane.showInputDialog(null, "Enter
the name of the astronaut");

```

```

        System.out.println("The number of missions for " +
name + " : " + ma.countMissions✓ (name) ✓);

```

```

    }

```

```

}

```