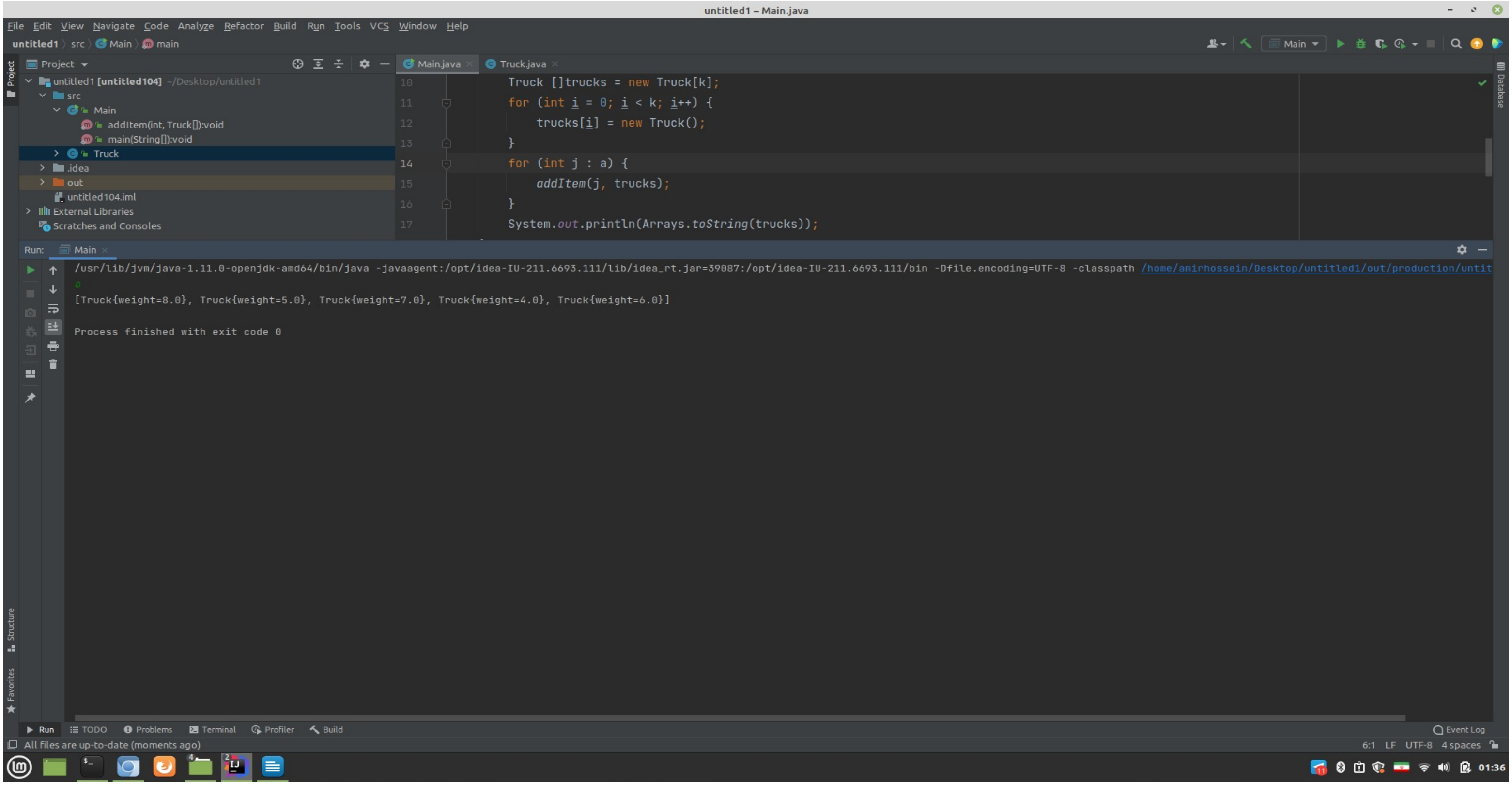
Project-1

برای حل پروژه اول الگوریتمی که به ذهن من رسید این بود که یک ارایه از وزن قطعات رو نگهداری میکنیم همچنین
یک ارایه از کامیون ها داریم که تعداد انها را از کاربر میگیریم سپس برای اضافه کردن هر قطعه به این کامیون ها به
این .شکل عمل میکنیم که میایم کم وزن ترین کامیون را پیدا کرده و قطعه را به این کامیون اضافه میکنیم

اجرای برنامه :

با تعداد ۵ کامیون


untitled1 – Main.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
untitled1 src Main main
Project
untitled1 [untitled104] ~/Desktop/untitled1
src
Main
addItem(int, Truck[]):void
main(String[]):void
Truck
.idea
out
untitled104.iml
External Libraries
Scratches and Consoles
Main.java
Truck.java
Run:
Main
Run TODO Problems Terminal Profiler Build
Event Log
All files are up-to-date (moments ago)
6:1 LF UTF-8 4 spaces
01:36

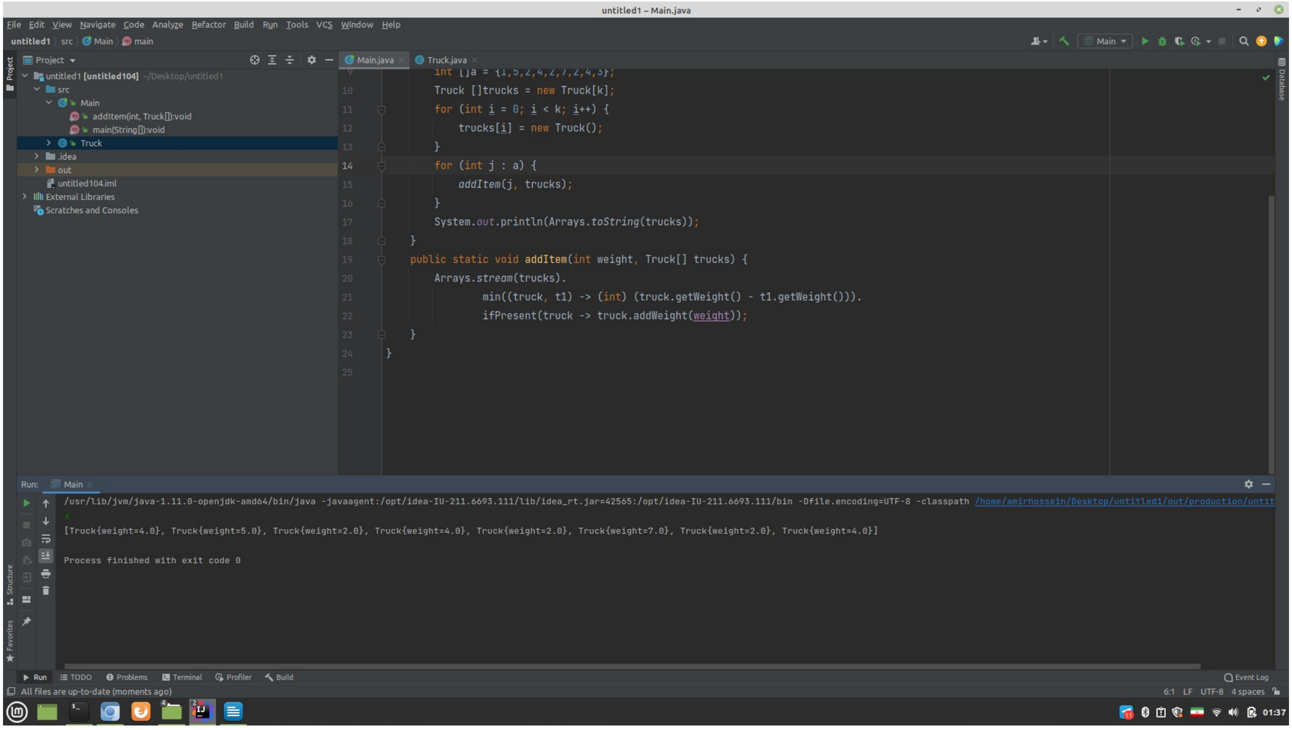
```java
        Truck []trucks = new Truck[k];
        for (int i = 0; i < k; i++) {
            trucks[i] = new Truck();
        }
        for (int j : a) {
            addItem(j, trucks);
        }
        System.out.println(Arrays.toString(trucks));
```

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/opt/idea-IU-211.6693.111/lib/idea_rt.jar=39087:/opt/idea-IU-211.6693.111/bin -Dfile.encoding=UTF-8 -classpath /home/amirhossein/Desktop/untitled1/out/production/untit

[Truck{weight=8.0}, Truck{weight=5.0}, Truck{weight=7.0}, Truck{weight=4.0}, Truck{weight=6.0}]

Process finished with exit code 0
```

با تعداد کامیون ۸:


untitled1 – Main.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
untitled1 src Main main
Project
untitled1 [untitled104] ~/Desktop/untitled1
src
Main
addItem(int, Truck[]):void
main(String[]):void
Truck
.idea
out
untitled104.iml
External Libraries
Scratches and Consoles
Main.java
Truck.java
Run: Main
Run TODO Problems Terminal Profiler Build
Event Log
All files are up-to-date (moments ago)
6:1 LF UTF-8 4 spaces
01:37

```java
        int []a = {1,5,2,4,2,7,2,4,3};
        Truck []trucks = new Truck[k];
        for (int i = 0; i < k; i++) {
            trucks[i] = new Truck();
        }
        for (int j : a) {
            addItem(j, trucks);
        }
        System.out.println(Arrays.toString(trucks));
    }
    public static void addItem(int weight, Truck[] trucks) {
        Arrays.stream(trucks).
                min((truck, t1) -> (int) (truck.getWeight() - t1.getWeight())).
                ifPresent(truck -> truck.addWeight(weight));
    }
}
```

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/opt/idea-IU-211.6693.111/lib/idea_rt.jar=42565:/opt/idea-IU-211.6693.111/bin -Dfile.encoding=UTF-8 -classpath /home/amirhossein/Desktop/untitled1/out/production/untit

[Truck{weight=4.0}, Truck{weight=5.0}, Truck{weight=2.0}, Truck{weight=4.0}, Truck{weight=2.0}, Truck{weight=7.0}, Truck{weight=2.0}, Truck{weight=4.0}]

Process finished with exit code 0
```