Your task is to create a class hierarchy like the one described below. The **BaseHero** class should be abstract.

* **BaseHero – string Name, int Power, string CastAbility()**
  + **Druid – power = 80**
  + **Paladin – power = 100**
  + **Rogue – power = 80**
  + **Warrior – power = 100**

Each hero should override the **CastAbility()** method:

**Druid - "{Type} - {Name} healed for {Power}"**

**Paladin - "{Type} - {Name} healed for {Power}"**

**Rogue - "{Type} - {Name} hit for {Power} damage"**

**Warrior - "{Type} - {Name} hit for {Power} damage"**

Now use the classes you created to form a raid group and defeat a boss. You will receive an integer **N** from the console. On the next lines, you will receive **{heroName}** and **{heroType}** until you create **N** number of heroes. If the hero type is invalid print: **"Invalid hero!"** and don’t add it to the raid group. After the raid group is formed you will receive an integer from the console which will be the boss’s power. Then each of the heroes in the raid group should cast his ability once. You should sum the power of all of the heroes and if the total power is greater or equal to the boss’s power you have defeated him and you should print:

**"Victory!"**

Else print:

**"Defeat..."**

### Bonus\*

Use the [Factory](https://www.c-sharpcorner.com/article/factory-method-design-pattern-in-c-sharp/) Design pattern to instantiate the classes.

### Constraints

You need to create heroes until you have **N** amount of **valid** heroes.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  Mike  Paladin  Josh  Druid  Scott  Warrior  250 | Paladin - Mike healed for 100  Druid - Josh healed for 80  Warrior - Scott hit for 100 damage  Victory! |
| 2  Mike  Warrior  Tom  Rogue  200 | Warrior - Mike hit for 100 damage  Rogue - Tom hit for 80 damage  Defeat... |