Problem 2. Job Offers

```
class JobOffers {
    //TODO Implement this class
}
```

Write a **class Job Offers**, which supports the described functionality below:

Functionality

Constructor

Should have these **3** properties:

- **`employer - string`**
- **`position - string`**
- **`jobCandidates - empty array`**

At the initialization of the **JobOffers** class, the **constructor** accepts the **employer** and **possition.**

Hint: You can add more properties to help you finish the task.

jobApplication(candidates)

This method adds people to the candidates list. The method takes **one argument**: **candidates (array of strings)**.

- **Every element** into this array is information about person in the format:

  `"{name}-{education}-{yearsExperience}"`

  - They are **separated** by slash symbol **`"-"`**. **Example**: [**`"John Doe-Bachelor-10", "Peter Parker-Master-5", "Daniel Jones- Bachelor-18"…`**]
  - If the **name** of the current person is already present in **`jobCandidates`** array, update the old **`yearsExperience`** **only if** the current one is **higher**.
  - Otherwise, should **add** the person, with properties: **`{name, education, yearsExperience}`** to the **`jobCandidates`** **array**.
- In **all cases**, you must **finally return a string** in the following format:

  `"You successfully added candidates: {name1}, {name2}, …{nameN}."`

  Note: When returning the **string**, keep in mind that the different **names** of **persons must** be:

  - **Unique** - for instance:
    - **`" You successfully added candidates: John Doe,Peter Parker,Daniel Jones "`** - is a correctly returned string
    - **`" You successfully added candidates: John Doe,Peter Parker,John Doe "`** - is not a correctly returned string
  - **Separated** by **comma** and **space (,)**

jobOffer(chosenPerson)

With this method, employer can sign contracts with person from the candidates list. The method takes **one argument**: **chosenPerson (string)**.

- The string about the selected player is in the format:

  `"{name}-{minimalExperience}"`

- Check:
  - If the **name** of the current person is not present in **`jobCandidates`** array, an **error** with the following message should be **thrown**:

    `"{name} is not in the candidates list!"`

  - If the **`minimalExperience`** selected by the employer for a given person **is more** than the value recorded in the array **`jobCandidates`**, an **error** with the following message should be **thrown**:

`"{name} does not have enough experience as {position}, minimum requirement is {minimalExperience} years."`

  - Otherwise, if the above conditions are not met, you must replace **`yearsExperience`** with the string "**hired**"
  - Finally, you need to **return** the string in the following format:

    `"Welcome aboard, our newest employee is {name}."`

salaryBonus(name)

With this method, we make sure that the candidates have **proper education (Bachelor degree or Master degree)** for the position. The method takes **one argument**:

  - **`name (string)`**

- If the submitted **name** is not present in the **`jobCandidates`** array, an **error** with the following message should be **thrown**:

  `"{name} is not in the candidates list!"`

- If the education recorded in the **`jobCandidates`** array is **Bachelor degree, return** the following message:

  `"{name} will sign a contract for {employer}, as {position} with a salary of $50,000 per year!"`

- If the education of the person is **Master degree, return** the following message:

  `"{name} will sign a contract for {employer}, as {position} with a salary of $60,000 per year!"`

- If the education recorded in the **`jobCandidates`** array is different than **Bachelor degree** or **Master degree**, **return** the following message:

"{name} will sign a contract for {employer}, as {position} with a salary of $40,000 per year!"

candidatesDatabase()

This method checks if there are any candidates in the database, if not **throw an error:**

"Candidate Database is empty!"

- Otherwise, **returns all candidates**, The first line shows the following message:

  "Candidates list:"

- On the new line, display information about each candidate sorted in **ascending** order of **name**:

  "{name}-{yearsExperience}"

Example

Input 1
`let Jobs = new JobOffers ("Google", "Strategy Analyst");` `console.log(Jobs.jobApplication(["John Doe-Bachelor-10", "Peter Parker-Master-5", "Daniel Jones- Bachelor-18"]));`

Output 1
You successfully added candidates: John Doe, Peter Parker, Daniel Jones.

Input 2

`let Jobs = new JobOffers ("Google", "Strategy Analyst");`
`console.log(Jobs.jobApplication(["John Doe-Bachelor-10", "Peter Parker-Master-5", "Daniel Jones- Bachelor-18"]));`
`console.log(Jobs.jobOffer("John Doe-8"));`
`console.log(Jobs.jobOffer("Peter Parker-4"));`
`console.log(Jobs.jobOffer("John Jones-8"));`

Output 2

```
You successfully added candidates: John Doe, Peter Parker, Daniel Jones.
Welcome aboard, our newest employee is John Doe.
Welcome aboard, our newest employee is Peter Parker.
Uncaught Error Error: John Jones is not in the candidates list!
```

Input 3

```
let Jobs = new JobOffers ("Google", "Strategy Analyst");
 console.log(Jobs.jobApplication(["John Doe-Bachelor-10", "Peter Parker-Master-5", "Daniel Jones- Bachelor-18"]));
 console.log(Jobs.jobOffer("John Doe-8"));
 console.log(Jobs.jobOffer("Peter Parker-4"));
 console.log(Jobs.salaryBonus("John Doe"));
 console.log(Jobs.salaryBonus("Peter Parker"));
```

Output 3

```
You successfully added candidates: John Doe, Peter Parker, Daniel Jones.
Welcome aboard, our newest employee is John Doe.
Welcome aboard, our newest employee is Peter Parker.
John Doe will sign a contract for Google, as Strategy Analyst with a salary of $50,000 per year!
Peter Parker will sign a contract for Google, as Strategy Analyst with a salary of $60,000 per year!
```

Input 4

```
let Jobs = new JobOffers ("Google", "Strategy Analyst");
 console.log(Jobs.jobApplication(["John Doe-Bachelor-10", "Peter Parker-Master-5","Jordan Cole-High School-5", "Daniel Jones- Bachelor-18"]));
 console.log(Jobs.jobOffer("John Doe-8"));
 console.log(Jobs.jobOffer("Peter Parker-4"));
 console.log(Jobs.jobOffer("Jordan Cole-4"));
 console.log(Jobs.salaryBonus("Jordan Cole"));
 console.log(Jobs.salaryBonus("John Doe"));
 console.log(Jobs.candidatesDatabase());
```

Output 4
You successfully added candidates: John Doe, Peter Parker, Jordan Cole, Daniel Jones.
Welcome aboard, our newest employee is John Doe.
Welcome aboard, our newest employee is Peter Parker.
Welcome aboard, our newest employee is Jordan Cole.
Jordan Cole will sign a contract for Google, as Strategy Analyst with a salary of $40,000 per year!
John Doe will sign a contract for Google, as Strategy Analyst with a salary of $50,000 per year!
Candidates list:
Daniel Jones-18
John Doe-hired
Jordan Cole-hired
Peter Parker-hired