DEPARTMENT (DIANA)

Salary above 65000

```
const employees = [
  { name: 'John', salary: 60000, department: 'HR' },
  { name: 'Jane', salary: 70000, department: 'IT' },
  { name: 'Mike', salary: 80000, department: 'IT' },
  { name: 'Sarah', salary: 55000, department: 'HR' },
  { name: 'David', salary: 75000, department: 'Finance' }
];

const averageSalaries = employees.reduce((acc, employee) => {
  if (!acc[employee.department]) {
    acc[employee.department] = { total: 0, count: 0 };
  }
  acc[employee.department].total += employee.salary;
  acc[employee.department].count++;
  return acc;
}, {});

const departmentsAbove65000 = Object.entries(averageSalaries)
  .map(([department, { total, count }]) => ({ department, averageSalary: total / count }))
  .filter(({ averageSalary }) => averageSalary > 65000)
  .map(({ department }) => department);

console.log(departmentsAbove65000);
```

Salary above 65000 (Different way)

```
const employees = [
  { name: 'John', salary: 60000, department: 'HR' },
  { name: 'Jane', salary: 70000, department: 'IT' },
  { name: 'Mike', salary: 80000, department: 'IT' },
  { name: 'Sarah', salary: 55000, department: 'HR' },
  { name: 'David', salary: 75000, department: 'Finance' }
];

const departmentSalaries = {};

employees.forEach(employee => {
  if (!departmentSalaries[employee.department]) {
    departmentSalaries[employee.department] = [];
  }
  departmentSalaries[employee.department].push(employee.salary);
});

const departmentsAbove65000 = Object.entries(departmentSalaries)
  .map(([department, salaries]) => ({
    department,
    averageSalary: salaries.reduce((acc, salary) => acc + salary, 0) / salaries.length
  }))
  .filter(({ averageSalary }) => averageSalary > 65000)
  .map(({ department }) => department);

console.log(departmentsAbove65000);
```