RISHITHA KIRAN, SARANYA MANOHARAN
4/12/17

TEAM 24 - MONGODB

1) Create a collection named "test", and insert into this collection the documents found in this link (10 documents): http://docs.mongodb.org/manual/reference/bios-example-collection/

```
var bulk = db.test.initializeUnorderedBulkOp();

bulk.insert({
   "_id" : 1,
   "name" : {
      "first" : "John",
      "last" : "Backus"
   },
   "birth" : ISODate("1924-12-03T05:00:00Z"),
   "death" : ISODate("2007-03-17T04:00:00Z"),
   "contribs" : [
      "Fortran",
      "ALGOL",
      "Backus-Naur Form",
      "FP"
   ],
   "awards" : [
      {
         "award" : "W.W. McDowell Award",
         "year" : 1967,
         "by" : "IEEE Computer Society"
      },
      {
         "award" : "National Medal of Science",
         "year" : 1975,
         "by" : "National Science Foundation"
      },
      {
         "award" : "Turing Award",
         "year" : 1977,
         "by" : "ACM"
      },
      {
         "award" : "Draper Prize",
         "year" : 1993,
         "by" : "National Academy of Engineering"
      }
   ]
});

bulk.insert({
   "_id" : ObjectId("51df07b094c6acd67e492f41"),
   "name" : {
      "first" : "John",
      "last" : "McCarthy"
   },
   "birth" : ISODate("1927-09-04T04:00:00Z"),
   "death" : ISODate("2011-12-24T05:00:00Z"),
   "contribs" : [
```

```
      "Lisp",
      "Artificial Intelligence",
      "ALGOL"
   ],
   "awards" : [
      {
         "award" : "Turing Award",
         "year" : 1971,
         "by" : "ACM"
      },
      {
         "award" : "Kyoto Prize",
         "year" : 1988,
         "by" : "Inamori Foundation"
      },
      {
         "award" : "National Medal of Science",
         "year" : 1990,
         "by" : "National Science Foundation"
      }
   ]
});

bulk.insert({
   "_id" : 3,
   "name" : {
      "first" : "Grace",
      "last" : "Hopper"
   },
   "title" : "Rear Admiral",
   "birth" : ISODate("1906-12-09T05:00:00Z"),
   "death" : ISODate("1992-01-01T05:00:00Z"),
   "contribs" : [
      "UNIVAC",
      "compiler",
      "FLOW-MATIC",
      "COBOL"
   ],
   "awards" : [
      {
         "award" : "Computer Sciences Man of the Year",
         "year" : 1969,
         "by" : "Data Processing Management Association"
      },
      {
         "award" : "Distinguished Fellow",
         "year" : 1973,
         "by" : " British Computer Society"
      },
      {
         "award" : "W. W. McDowell Award",
         "year" : 1976,
         "by" : "IEEE Computer Society"
```

```
        },
        {
            "award" : "National Medal of Technology",
            "year" : 1991,
            "by" : "United States"
        }
    ]
});

bulk.insert({
    "_id" : 4,
    "name" : {
        "first" : "Kristen",
        "last" : "Nygaard"
    },
    "birth" : ISODate("1926-08-27T04:00:00Z"),
    "death" : ISODate("2002-08-10T04:00:00Z"),
    "contribs" : [
        "OOP",
        "Simula"
    ],
    "awards" : [
        {
            "award" : "Rosing Prize",
            "year" : 1999,
            "by" : "Norwegian Data Association"
        },
        {
            "award" : "Turing Award",
            "year" : 2001,
            "by" : "ACM"
        },
        {
            "award" : "IEEE John von Neumann Medal",
            "year" : 2001,
            "by" : "IEEE"
        }
    ]
});

bulk.insert({
    "_id" : 5,
    "name" : {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    },
    "birth" : ISODate("1931-10-12T04:00:00Z"),
    "death" : ISODate("2002-06-29T04:00:00Z"),
    "contribs" : [
        "OOP",
        "Simula"
    ],
    "awards" : [
```

```
        {
            "award" : "Rosing Prize",
            "year" : 1999,
            "by" : "Norwegian Data Association"
        },
        {
            "award" : "Turing Award",
            "year" : 2001,
            "by" : "ACM"
        },
        {
            "award" : "IEEE John von Neumann Medal",
            "year" : 2001,
            "by" : "IEEE"
        }
    ]
});

bulk.insert({
    "_id" : 6,
    "name" : {
        "first" : "Guido",
        "last" : "van Rossum"
    },
    "birth" : ISODate("1956-01-31T05:00:00Z"),
    "contribs" : [
        "Python"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : 2001,
            "by" : "Free Software Foundation"
        },
        {
            "award" : "NLUUG Award",
            "year" : 2003,
            "by" : "NLUUG"
        }
    ]
});

bulk.insert({
    "_id" : ObjectId("51e062189c6ae665454e301d"),
    "name" : {
        "first" : "Dennis",
        "last" : "Ritchie"
    },
    "birth" : ISODate("1941-09-09T04:00:00Z"),
    "death" : ISODate("2011-10-12T04:00:00Z"),
    "contribs" : [
        "UNIX",
        "C"
```

```
    ],
    "awards" : [
        {
            "award" : "Turing Award",
            "year" : 1983,
            "by" : "ACM"
        },
        {
            "award" : "National Medal of Technology",
            "year" : 1998,
            "by" : "United States"
        },
        {
            "award" : "Japan Prize",
            "year" : 2011,
            "by" : "The Japan Prize Foundation"
        }
    ]
});

bulk.insert({
    "_id" : 8,
    "name" : {
        "first" : "Yukihiro",
        "aka" : "Matz",
        "last" : "Matsumoto"
    },
    "birth" : ISODate("1965-04-14T04:00:00Z"),
    "contribs" : [
        "Ruby"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : "2011",
            "by" : "Free Software Foundation"
        }
    ]
});

bulk.insert({
    "_id" : 9,
    "name" : {
        "first" : "James",
        "last" : "Gosling"
    },
    "birth" : ISODate("1955-05-19T04:00:00Z"),
    "contribs" : [
        "Java"
    ],
    "awards" : [
        {
            "award" : "The Economist Innovation Award",
```

```
            "year" : 2002,
            "by" : "The Economist"
        },
        {
            "award" : "Officer of the Order of Canada",
            "year" : 2007,
            "by" : "Canada"
        }
    ]
});

bulk.insert({
    "_id" : 10,
    "name" : {
        "first" : "Martin",
        "last" : "Odersky"
    },
    "contribs" : [
        "Scala"
    ]
});

bulk.execute();
```

OUTPUT:

```
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 10,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
```

---

2) Write a CRUD operation(s) that changes the _id of "John McCarthy" to value 2.

```
changeID = db.test.findOne({"name.first" : "John", "name.last" : "McCarthy" } );
changeID._id = 2;
db.test.insert(changeID);
db.test.remove({"name.first" : "John", "name.last" : "McCarthy", "_id" : {$ne: 2} })
```

OUTPUT:

```
> changeID._id = 2;
2
> db.test.insert(changeID);
WriteResult({ "nInserted" : 1 })
> db.test.remove({"name.first" : "John", "name.last" : "McCarthy", "_id" : {$ne: 2} })
WriteResult({ "nRemoved" : 1 })
```

WHEN TRIED USING UPDATE FUNCTION:

Since "_id" is the primary key update function failed when we executed the below query:

```
> db.test.update({ "name.first": "John" , "name.last": "McCarthy" },{$set: { "_id" :2 } },{multi: true } );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
```

---

3) Write a CRUD operation(s) that inserts the following new records into the collection:

```
db.test.insert({
        "_id" : 20,
        "name" :
        {
                "first" : "Alex",
                "last" : "Chen"
        },
        "birth" : ISODate("1933-08-27T04:00:00Z"),
        "death" : ISODate("1984-11-07T04:00:00Z"),
        "contribs" :
        [
                "C++",
                "Simula"
        ],
        "awards" :
        [
                {
                        "award" : "WPI Award",
                        "year" : 1977,
                        "by" : "WPI"
                }
        ]
});
```

OUTPUT:

```
WriteResult({ "nInserted" : 1 })
```

```
db.test.insert({
        "_id" : 30,
        "name" :
        {
                "first" : "David",
                "last" : "Mark"
        },
        "birth" : ISODate("1911-04-12T04:00:00Z"),
        "death" : ISODate("2000-11-07T04:00:00Z"),
        "contribs" :
        [
                "C++",
                "FP",
                "Lisp",
        ],
        "awards" :
        [
                {
                        "award" : "WPI Award",
                        "year" : 1963,
```

```
                "by" : "WPI"
            },
            {
                "award" : "Turing Award",
                "year" : 1966,
                "by" : "ACM"
            }
        ]
});
```

OUTPUT:

```
WriteResult({ "nInserted" : 1 })
```

---

4) Report all documents of people who got a "Turing Award" after 1976

```
db.test.find({"awards":{$elemMatch:{"award" : "Turing Award" ,"year" : { $gt : 1976 }}}});
```

OUTPUT:

```
{ "_id" : 1, "name" : { "first" : "John", "last" : "Backus" }, "birth" : ISODate("1924-12-03T05:00:00Z"), "death" : ISODate("2007-03-17T04:00:00Z"), "contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ], "awards" : [ { "award" : "W.W. McDowell Award", "year" : 1967, "by" : "IEEE Computer Society" }, { "award" : "National Medal of Science", "year" : 1975, "by" : "National Science Foundation" }, { "award" : "Turing Award", "year" : 1977, "by" : "ACM" }, { "award" : "Draper Prize", "year" : 1993, "by" : "National Academy of Engineering" } ] }
{ "_id" : 4, "name" : { "first" : "Kristen", "last" : "Nygaard" }, "birth" : ISODate("1926-08-27T04:00:00Z"), "death" : ISODate("2002-08-10T04:00:00Z"), "contribs" : [ "OOP", "Simula" ], "awards" : [ { "award" : "Rosing Prize", "year" : 1999, "by" : "Norwegian Data Association" }, { "award" : "Turing Award", "year" : 2001, "by" : "ACM" }, { "award" : "IEEE John von Neumann Medal", "year" : 2001, "by" : "IEEE" } ] }
{ "_id" : 5, "name" : { "first" : "Ole-Johan", "last" : "Dahl" }, "birth" : ISODate("1931-10-12T04:00:00Z"), "death" : ISODate("2002-06-29T04:00:00Z"), "contribs" : [ "OOP", "Simula" ], "awards" : [ { "award" : "Rosing Prize", "year" : 1999, "by" : "Norwegian Data Association" }, { "award" : "Turing Award", "year" : 2001, "by" : "ACM" }, { "award" : "IEEE John von Neumann Medal", "year" : 2001, "by" : "IEEE" } ] }
{ "_id" : ObjectId("51e062189c6ae665454e301d"), "name" : { "first" : "Dennis", "last" : "Ritchie" }, "birth" : ISODate("1941-09-09T04:00:00Z"), "death" : ISODate("2011-10-12T04:00:00Z"), "contribs" : [ "UNIX", "C" ], "awards" : [ { "award" : "Turing Award", "year" : 1983, "by" : "ACM" }, { "award" : "National Medal of Technology", "year" : 1998, "by" : "United States" }, { "award" : "Japan Prize", "year" : 2011, "by" : "The Japan Prize Foundation" } ] }
```

---

5) Report all documents of people who got less than 3 awards or have contribution in "FP"

```
db.test.find({$or: [ {"contribs" : "FP"}, {"awards" : {$exists: true} , $where: "this.awards.length < 3" },{"awards" : {$exists: false}}]});
```

OUTPUT:

```
{ "_id" : 1, "name" : { "first" : "John", "last" : "Backus" }, "birth" : ISODate("1924-12-03T05:00:00Z"), "contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ], "awards" : [ { "award" : "W.W. McDowell Award", "year" : 1967, "by" : "IEEE Computer Society" }, { "award" : "National Medal of Science", "year" : 1975, "by" : "National Science Foundation" }, { "award" : "Turing Award", "year" : 1977, "by" : "ACM" }, { "award" : "Draper Prize", "year" : 1993, "by" : "National Academy of Engineering" } ] }
{ "_id" : 6, "name" : { "first" : "Guido", "last" : "van Rossum" }, "birth" : ISODate("1956-01-31T05:00:00Z"), "contribs" : [ "Python", "OOP" ], "awards" : [ { "award" : "Award for the Advancement of Free Software", "year" : 2001, "by" : "Free Software Foundation" }, { "award" : "NLUUG Award", "year" : 2003, "by" : "NLUUG" } ] }
```

{ "_id" : ObjectId("51e062189c6ae665454e301d"), "name" : { "first" : "Dennis", "last" : "Ritchie" }, "birth" : ISODate("1941-09-09T04:00:00Z"), "contribs" : ["UNIX", "C"], "awards" : [{ "award" : "Turing Award", "year" : 1983, "by" : "ACM" }, { "award" : "National Medal of Technology", "year" : 1998, "by" : "United States" }] }
{ "_id" : 8, "name" : { "first" : "Yukihiro", "aka" : "Matz", "last" : "Matsumoto" }, "birth" : ISODate("1965-04-14T04:00:00Z"), "contribs" : ["Ruby"], "awards" : [ ] }
{ "_id" : 9, "name" : { "first" : "James", "last" : "Gosling" }, "birth" : ISODate("1955-05-19T04:00:00Z"), "contribs" : ["Java"], "awards" : [{ "award" : "The Economist Innovation Award", "year" : 2002, "by" : "The Economist" }, { "award" : "Officer of the Order of Canada", "year" : 2007, "by" : "Canada" }] }
{ "_id" : 10, "name" : { "first" : "Martin", "last" : "Odersky" }, "contribs" : ["Scala"] }
{ "_id" : 20, "name" : { "first" : "Alex", "last" : "Chen" }, "birth" : ISODate("1933-08-27T04:00:00Z"), "contribs" : ["C++", "Simula"], "awards" : [{ "award" : "WPI Award", "year" : 1977, "by" : "WPI" }], "comments" : ["He taught in 3 universities", "died from cancer", "lived in CA"] }
{ "_id" : 30, "name" : { "first" : "David", "last" : "Mark" }, "birth" : ISODate("1911-04-12T04:00:00Z"), "death" : ISODate("2000-11-07T04:00:00Z"), "contribs" : ["C++", "FP", "Lisp", "UNIVAC", "compiler", "FLOW-MATIC", "COBOL"], "awards" : [{ "award" : "WPI Award", "year" : 1963, "by" : "WPI" }, { "award" : "Turing Award", "year" : 1966, "by" : "ACM" }] }

---

6) Report the contributions of "Dennis Ritchie" (only report the name and the contribution array)

```
db.test.find({"name.first": "Dennis", "name.last": "Ritchie"},{"name": 1 ,"contribs": 1, "_id": 0});
```

OUTPUT:

{ "name" : { "first" : "Dennis", "last" : "Ritchie" }, "contribs" : ["UNIX", "C"] }

---

7) Update the document of "Guido van Rossum" to add "OOP" to the contribution list.

```
db.test.update({"name.first": "Guido", "name.last" : "van Rossum"},{$push: {"contribs": "OOP"}});
```

OUTPUT:

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.test.find({"name.first":"Guido"});
{ "_id" : 6, "name" : { "first" : "Guido", "last" : "van Rossum" }, "birth" : ISODate("1956-01-31T05:00:00Z"), "contribs" : ["Python", **"OOP"**], "awards" : [{ "award" : "Award for the Advancement of Free Software", "year" : 2001, "by" : "Free Software Foundation" }, { "award" : "NLUUG Award", "year" : 2003, "by" : "NLUUG" }] }

---

8) Insert a new filed of type array, called "comments", into the document of "Alex Chen" storing the following comments: "He taught in 3 universities", "died from cancer", "lived in CA"

```
db.test.update({"name.first": "Alex", "name.last": "Chen"},
        {$set: {"comments": ["He taught in 3 universities", "died from cancer", "lived in CA"]}},{upsert: true});
```

OUTPUT:

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.test.find({"name.first":"Alex"});
{ "_id" : 20, "name" : { "first" : "Alex", "last" : "Chen" }, "birth" : ISODate("1933-08-27T04:00:00Z"), "death" : ISODate("1984-11-07T04:00:00Z"), "contribs" : ["C++", "Simula"], "awards" : [{ "award" : "WPI Award", "year" : 1977, "by" : "WPI" }], **"comments" : ["He taught in 3 universities", "died from cancer", "lived in CA"]** }

---

9) For each contribution by "Alex Chen", say X, list the peoples' names (first and last) who have contribution X. E.g., Alex Chen has two contributions in "C++" and "Simula". Then, the output should be similar to:

a. {Contribution: "C++", People: [{first: "Alex", last: "Chen"}, {first: "David", last: "Mark"}]},
{ Contribution: "Simula",….}

```
db.test.find({"name.first": "Alex", "name.last": "Chen"}).forEach(function(item)
{
        con = item.contribs;
        con.forEach(function(doc)
        {
                print("Contributions: " +doc+ ",\nPeople: ")
                var people = [];
                db.test.find({"contribs": doc}).forEach( function(n)
                {
                        people.push({first: n.name.first, last: n.name.last})
                });
                printjson(people)
        });
});
```

OUTPUT:

```
Contributions: C++,
People:
[
    {
        "first" : "Alex",
        "last" : "Chen"
    },
    {
        "first" : "David",
        "last" : "Mark"
    }
]
Contributions: Simula,
People:
[
    {
        "first" : "Kristen",
        "last" : "Nygaard"
    },
    {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    },
    {
        "first" : "Alex",
        "last" : "Chen"
    }
]
```

---

10) Report all documents where the first name matches the regular expression "Jo*", where "*" means any number of characters. Report the documents sorted by the last name.

```
db.test.find({"name.first": { $regex: /^Jo/}}).sort({"name.last": 1});
```

OUTPUT:

{ "_id" : 1, "name" : { "first" : "John", "last" : "Backus" }, "birth" : ISODate("1924-12-03T05:00:00Z"), "death" : ISODate("2007-03-17T04:00:00Z"), "contribs" : ["Fortran", "ALGOL", "Backus-Naur Form", "FP"], "awards" : [{ "award" : "W.W. McDowell Award", "year" : 1967, "by" : "IEEE Computer Society" }, { "award" : "National Medal of Science", "year" : 1975, "by" : "National Science Foundation" }, { "award" : "Turing Award", "year" : 1977, "by" : "ACM" }, { "award" : "Draper Prize", "year" : 1993, "by" : "National Academy of Engineering" }] }
{ "_id" : 2, "name" : { "first" : "John", "last" : "McCarthy" }, "birth" : ISODate("1927-09-04T04:00:00Z"), "death" : ISODate("2011-12-24T05:00:00Z"), "contribs" : ["Lisp", "Artificial Intelligence", "ALGOL"], "awards" : [{ "award" : "Turing Award", "year" : 1971, "by" : "ACM" }, { "award" : "Kyoto Prize", "year" : 1988, "by" : "Inamori Foundation" }, { "award" : "National Medal of Science", "year" : 1990, "by" : "National Science Foundation" }] }

---

11) Report the distinct organization that gave awards. This information can be found in the "by" field inside the "awards" array. The output should be an array of the distinct values, e.g., ["wpi', "acm', …]

```
db.test.distinct("awards.by");
```

OUTPUT:

```
[
 "ACM",
 "IEEE Computer Society",
 "National Academy of Engineering",
 "National Science Foundation",
 " British Computer Society",
 "Data Processing Management Association",
 "United States",
 "IEEE",
 "Norwegian Data Association",
 "Free Software Foundation",
 "NLUUG",
 "The Japan Prize Foundation",
 "Canada",
 "The Economist",
 "Inamori Foundation",
 "WPI"
]
```

---

12) Delete from all documents the "death" field.

```
db.test.update({"death":{$exists : true}}, {$unset : {death : ""}}, {multi : true});
```

OUTPUT:

```
WriteResult({ "nMatched" : 8, "nUpserted" : 0, "nModified" : 8 })
```

---

13) Delete from all documents any award given on 2011.

```
db.test.update({}, {$pull: { "awards": {$or: [{"year": { $eq: "2011" }},{"year": { $eq: 2011 }}]}}},{ multi: true });
```

OUTPUT:

```
WriteResult({ "nMatched" : 12, "nUpserted" : 0, "nModified" : 2 })
```

---

14) Update the award of document _id =30, which is given by WPI, and set the year to 1965.

```
db.test.update({"_id": 30, "awards.by": "WPI"},{$set: {"awards.$.year": 1965}});
```

OUTPUT:
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

---

15) Add (copy) all the contributions of document _id = 3 to that of document _id = 30

```
db.test.find({"_id":3}).forEach(function(input)
{
        value = input.contribs;
        value.forEach(function(v)
        {
                db.test.update({"_id":30},{$push: {"contribs": v}})
        });
});
```

OUTPUT:
> db.test.find({"_id":30});
{ "_id" : 30, "name" : { "first" : "David", "last" : "Mark" }, "birth" : ISODate("1911-04-12T04:00:00Z"), "death" : ISODate("2000-11-07T04:00:00Z"), "contribs" : ["C++", "FP", "Lisp", **"UNIVAC", "compiler", "FLOW-MATIC", "COBOL"**], "awards" : [{ "award" : "WPI Award", "year" : 1963, "by" : "WPI" }, { "award" : "Turing Award", "year" : 1966, "by" : "ACM" }] }

> db.test.find({"_id":3});
{ "_id" : 3, "name" : { "first" : "Grace", "last" : "Hopper" }, "title" : "Rear Admiral", "birth" : ISODate("1906-12-09T05:00:00Z"), "contribs" : [**"UNIVAC", "compiler", "FLOW-MATIC", "COBOL"**], "awards" : [{ "award" : "Computer Sciences Man of the Year", "year" : 1969, "by" : "Data Processing Management Association" }, { "award" : "Distinguished Fellow", "year" : 1973, "by" : " British Computer Society" }, { "award" : "W. W. McDowell Award", "year" : 1976, "by" : "IEEE Computer Society" }, { "award" : "National Medal of Technology", "year" : 1991, "by" : "United States" }] }

---

16) Report only the names (first and last) of those individuals who won at least two awards in 2001.

```
db.test.find({"awards": {$exists: true} , $where: "this.awards.length > 2","awards.year": 2001} , {"name.first" : 1 , "name.last":1,"_id":0})
```

OUTPUT:
{ "name" : { "first" : "Kristen", "last" : "Nygaard" } }
{ "name" : { "first" : "Ole-Johan", "last" : "Dahl" } }

---

17) Report the document with the largest id. First, you need to find the largest _id (using a CRUD statement), and then use that to report the corresponding document.

```
db.test.find({}).sort({"_id": -1}).limit(1);
```

OUTPUT:
{ "_id" : ObjectId("51e062189c6ae665454e301d"), "name" : { "first" : "Dennis", "last" : "Ritchie" }, "birth" : ISODate("1941-09-09T04:00:00Z"), "contribs" : ["UNIX", "C"], "awards" : [{ "award" : "Turing Award", "year" : 1983, "by" : "ACM" }, { "award" : "National Medal of Technology", "year" : 1998, "by" : "United States" }] }

---

18) Report only one document where one of the awards is given by "ACM".

```
db.test.findOne({"awards.by": "ACM"});
```

OR

db.test.find({ "awards.by": "ACM" }).limit(1).pretty();

OUTPUT:

{ "_id" : 1, "name" : { "first" : "John", "last" : "Backus" }, "birth" : ISODate("1924-12-03T05:00:00Z"), "contribs" : ["Fortran", "ALGOL", "Backus-Naur Form", "FP"], "awards" : [{ "award" : "W.W. McDowell Award", "year" : 1967, "by" : "IEEE Computer Society" }, { "award" : "National Medal of Science", "year" : 1975, "by" : "National Science Foundation" }, { "award" : "Turing Award", "year" : 1977, "by" : "ACM" }, { "award" : "Draper Prize", "year" : 1993, "by" : "National Academy of Engineering" }] }

---

19) Delete the documents inserted in Q3, i.e., _id = 20 and 30.

db.test.remove({$or: [{"_id": 20},{"_id":30}]});

OR

db.test.remove({"_id": {$in: [20 , 30] } });

OUTPUT:

WriteResult({ "nRemoved" : 2 })

---

20) Report the number of documents in the collection.

db.test.count();

OUTPUT:

> db.test.count();
10
