Homework8

1.

i) ptr1 is assigned to ptr2, so ptr1 and ptr2 points to the same location which is x, so the value of the pointers are the same. When \*ptr1 = 9 is executed, the value of x is set to 9, so the value of \*ptr1 and \*ptr2 are both 9.

l) Similar to what is discussed in i. When \*ptr2 = 15 is executed, the value of x is set to 15, so the value of \*ptr1 and ptr2 are both 15. And the value of the pointers remains the same.

q) When ptr1 = arr is executed, the pointer ptr1 points to the array arr but pointer ptr2 remains pointing to x. So the value of ptr2 and \*ptr2 will not change. When ptr1 = arr is executed, the pointer ptr1 actually points to the first element of array arr which is arr[0]. So the value of \*ptr1 is 4. When ptr1++ is executed, the pointer ptr1 then points to the next element of the array arr, which is arr[1]. So the value of \*ptr1 is 12, and so on.

2.

**Cluster:** A set of Nodes (servers) that holds all the data.

**Node:** A single server that holds some data and participate on the cluster's indexing and querying.

**Index:** Forget SQL Indexes. Each ES Index is a set of Documents.

**Thread pools:** A thread pool is a collection of worker threads that efficiently execute asynchronous callbacks on behalf of the application.

**Shards:** A subset of Documents of an Index. An Index can be divided in many shards.

**Type:** A definition of the schema of a Document inside of an Index (a Index can have more than one type assigned).

**Document:** A JSON object with some data. It's the basic information unit in ES.

**ES uses threads to delete, update, insert (True/False)? How does it work?**

TRUE：

Insert: data-> buffer-> index segment-> os cache (commit)-> os disk. At this time, the buffer will be emptied, and the web end can search the data just inserted.

Delete: Each time a commit is made, the file will be marked as .del. When you search again, you can see that the data in the segment is in the .del state and it will not be queried.

Update: Mark the existing file as .del, then write the new document file into the new index segment.

**How does ES handles when a Node in ES cluster goes down?**

Enter curl 'http:// ip: port/\_ nodes/ process? Pretty' to view the specific information of the nodes in the es cluster. Enter curl -XGET http: // ip: port / \_cat / shards | grep UNASSIGNED to see the information (index, id, etc.) in the unassigned state. Finally force routing rerouting.

**ElasticSearch use REST architecture (Yes/No)?What is Rest Architecture?**

Yes:

Representational State Transfer (REST) is an architectural style for creating, maintaining, retrieving, and deleting resources. REST's information-driven, resource-oriented approach to building Web services can both satisfy your software's users and make your life as a developer easier.

**Spring MVC? How does it work?**

Spring MVC:

Spring Web MVC is the original web framework built on the Servlet API and has been included in the Spring Framework from the very beginning.

Working Principle:

It all starts with the client, which sends a request to a specific URL. When that request hits the web container e.g. Tomcat it look into web.xml and find the Servlet or Filter which is mapped to that particular URL. It the delegate that Servlet or Filter to process the request. Since Spring MVC is built on top of Servlet.

**Is ElasticSearch structured database or unstructured database? Give an example**

Elasticsearch is a structured Database.

Example: An ES can contain multiple indexes (databases), each index contains many types (tables), the types contain many documents (rows), each document uses JSON format to store data, and contains many partitions (columns)

**What is document-oriented database? Give an example**

A document-oriented database is a computer program designed for storing, retrieving and managing document-oriented information. Document-oriented databases are inherently a subclass of the key-value store, another NoSQL database concept.

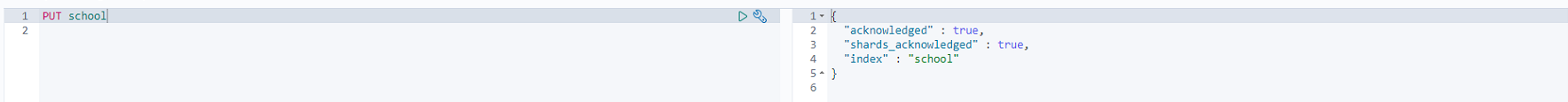
Example: Apache CouchDB is a document-oriented database management system.

**What is Kabina?**

Kabina is GUI interface to Elasticsearch engine. A normal user can use kabina to do send/receive various operations to elasticsearch.

3.

(a) is presented in the Java code.







4.

A pointer to a character: char\* ptr1;

An array of 10 integers: int arr1[10];

A reference to an array of 10 integers: int (&a)[10] = arr1;

A pointer to an array of character strings: char arr2[4] = {'a', 'b', 'c', 'd'};

char\* ptr2 = arr2;

A pointer to a pointer to a character: char\* ptr3 = ptr1;

A constant integer: const int b;

A pointer to a constant integer: int\* const ptr4;

Constant pointer to an integer: const int\* ptr5;

8.

in this example the same function is used in each thread. The arguments are different. The functions need not be the same.

Threads terminate by explicitly calling pthread\_exit(), by letting the function return, or by a call to the function exit() which will terminate the process including any threads.

Function call: pthread\_create - create a new thread

Function call: pthread\_join - wait for termination of another thread

Function call: pthread\_join - wait for termination of another thread