**Yelp**

1. **What happens when you type "yelp.com" on your browser**

You send a DNS packet (a UDP packet) to your DNS server, getting the mapping of the domain name "yelp.com" to an IP address. The MAC address of the link layer header of this packet is your gateway router, which then forwards this packet to your DNS server.

If your DNS server doesn't have a cached entry for "yelp.com", it will have to iteratively query the root nameserver, TLDs, and yelp's nameserver in order to get the mapping, which it then returns to you.

Now, you need to complete the TCP three-way handshake in order to establish a connection between your computer and yelp.com you send a TCP SYN packet, wait for a SYN ACK, then send an ACK packet.

Now you can get the content of "yelp.com". You send a HTTP GET request (under the TCP protocol), and wait for the HTTP responses from yelp.com. You may get several packets due to TCP segmentation.

----------------------------------------------------------------**or**-------------------------------------------------------------

On a higher level, you could talk about browser cookies, and the fact that the yelp.com server will (1) run some code like python to find information about your profile (2) which will eventually call into some database to get some information like how many friends, reviews, does this person have, what’s his favorite location, etc. (3) maybe to get your profile photo (if not already cached by the browser) will talk to some key value store to get photos, (like amazon AWS)

Then you can ask the interviewer which part would he like more details on... I think it’s meant to be a conversation.

1. **Check anagram**
2. **C++ program to check if given strings are anagrams**

int find\_anagram(string array1, string array2)

{

int num1[26] = { 0 }, i = 0;

while (array1[i] != 0)

{

num1[tolower(array1[i]) - 'a']++;

// example: let array1[i] is 'A' =>

tolower(A) = 'a' =>

num1[ascii('a') - ascii('a')] =>

num1[0]++ =>

num1[0] is count of 'a'

i++;

}

i = 0;

while (array2[i] != 0)

{

num1[tolower(array2[i]) - 'a']--;

i++;

}

for (i = 0; i < 26; i++)

{

if (num1[i] != 0)

return false;

}

return true;

}

1. **C++ program to print only the anagrams in the given list**

int listAnagram()

{

string input[] = {"cat", "odg" ,"tac", "dog" ,"atc", "abc"};

int len = sizeof(input) / sizeof(input[0]);

vector<string> ip (input,input+len);

map<string, vector<int>> strmap;

int count = 0;

for (vector<string>::iterator i = ip.begin(); i < ip.end(); i++)

{

sort((\*i).begin(),(\*i).end());

map<string, vector<int> > ::iterator vi = strmap.find(\*i);

if (vi == strmap.end())

{

vector<int> ct;

ct.push\_back(count);

strmap[\*i] = ct;

}

else

{

vector<int> ct = vi->second;

ct.push\_back(count);

strmap[\*i] = ct;

}

count++;

}

vector<string> finalList;

for (map<string, vector<int>>::iterator it = strmap.begin(); it != strmap.end(); it++)

{

//vector<int> fList = it->second;

if (it->second.size() > 1)

{

for (vector<int>::iterator itr = it->second.begin(); itr != it->second.end(); itr++)

{

finalList.push\_back(input[\*itr]);

}

}

}

return 0;

}

/\*

---------------

|\_map\_\_vector\_|

|act |{0,2,4} |

|dog |{1,5} |

---------------

\*/

1. **Reverse a string (C++)**

using namespace std;

#include <iostream>

class reverseString

{

private:

char\* getStr;

void swap(char \*a, char\* b)

{

char temp = \*a;

\*a = \*b;

\*b = temp;

}

public:

reverseString(char\* ip)

{

getStr = ip;

}

void reverse()

{

int len = strlen(getStr);

for (int i = 0; i < len / 2; i++)

{

swap(&getStr[i], &getStr[len - i - 1]);

}

}

};

int main()

{

char data[] = "check";

reverseString \*rev = new reverseString(data);

rev->reverse();

cout << data;

return 0;

}

1. How to improve performance of websites
   1. Minimize HTTP Requests: Simplify your page design.
   2. Reduce server response time:
   3. Enable compression: Compression reduces the bandwidth of your pages, thereby reducing HTTP response. You do this with a tool called Gzip. Most web servers can compress files in Gzip format before sending them for download, either by calling a third-party module or using built-in routines.
   4. Enable browser caching
   5. Optimize images
   6. Optimize css Delivery
2. Why my website is slow?
   1. A huge spike in Additional Traffic.
   2. A Dynamic Site without caching.
   3. Extremely Large or non-optimized graphics.
   4. Extremely Large or non-optimized database.
   5. Large amount of calls to external websites (twitter, facebook, youtube, etc...)
   6. Connectivity issues.

**Interview**

1. Why Yelp?
2. About my project, I had worked on yelp dataset challenge.
3. About my distributed systems project.
4. What happens when I request a Server?
   1. I said http request.
5. What protocol does it uses.
   1. Tcp/Ip
6. How does Tcp/Ip work ?
7. I mentioned about SSL/TLS, so he asked me why that? What’s the use?
8. I mentioned encryption. He asked, what type of encryption?
9. Then a coder pad:
   1. I have a list of classes and class schedule.
      1. cs100 : Mon-Tue: 2-3 , Wed-Thur: 4-5
      2. cs111 : Mon-Tue: 4-5 , Wed-Thur: 4-5
      3. cs123 : Mon-Tue: 2-3 , Wed-Thur: 7-5
   2. He asked me how will you find which course will be best suitable for a student, without collision of timings.

My Answer: a course hash map <course, list<class\_schedule>> and another hash map <class\_schedule, list<courses>>. Take the class\_schedule which has only one course, as that will be first chosen. Then iterate through the course hash map from left to right and consider each one and block the schedule for that (Random pick). **\*\* If there are no preferences given**.