**1. Describe yourself**

This is Priya Singh, I would describe myself as highly motivated, hard-working, self-driven and particularly enjoy working as a part of team, in my current job I am part of a group of 12 people and we communicate multiple time to work as unit. I enjoy fast-paced, team-oriented environment like this. I know when to ask for help and don’t quite stay if I do need any assistance but If it is possible to handle Without occupying the time of other I do it and consider myself very good at it. I have done my BE from Computer Science Stream with 76%, currently I am working as Senior Software Engineer in Happiest Minds Tech Pvt. Ltd with overall experience is 3.3 years including my internship. I am working on Frontend development, having good technical exposure in JavaScript, AngularJs, Html, CSS, Bootstrap. Having basic knowledge of Elastic Search, NodeJS, Angular2, Automation, Unit testing, Manual testing.

**2.You discover a bug in the codebase. What steps do you take?**

If I discover any bug in codebase I will follow these steps

Step1. I make sure truly understand the problem.

Step2. I debug the code with the help of source map and dev-tools in browser.

Step3. If I am not able to solve it, then I google the error message.

Step4. Identify the immediate line of code where the bug occurs.

Step5. I will identify the species of bug.

Step6. I log the specific line of code and analyze the log message.

Step7. Fix the bug.

Step8. Test all the scenario of producing the bug.

Step9. Get a second pair of eyes.

Step10. Prove the fix and deliver it.

**3.Describe the kinds of tests you would write at various points in the SDLC**

SDLC is stand for “Software Development Life Cycle” this is process which is followed by software Industry to design, develop and test high quality software. The SDLC aims to produce high quality software to meet and exceeds the customer expectations, reaches completion within time and cost estimates.

The SDLC define a methodology to improving the quality of software and overall development process. The SDLC highlights different stages (phrases or steps) of the development process.

### **1. Planning**

This is the first phase in the systems development process. It identifies whether or not there is the need for a new system to achieve a business’s strategic objectives. This is a preliminary plan (or a feasibility study) for a company’s business initiative to acquire the resources to build on an infrastructure to modify or improve a service. The company might be trying to meet or exceed expectations for their employees, customers and stakeholders too. The purpose of this step is to find out the scope of the problem and determine solutions. Resources, costs, time, benefits and other items should be considered at this stage.

### **2. Systems Analysis and Requirements**

The second phase is where businesses will work on the source of their problem or the need for a change. In the event of a problem, possible solutions are submitted and analyzed to identify the best fit for the ultimate goal(s) of the project. This is where teams consider the functional requirements of the project or solution. It is also where system analysis takes place—or analyzing the needs of the end users to ensure the new system can meet their expectations. Systems analysis is vital in determining what a business’s needs are, as well as how they can be met, who will be responsible for individual pieces of the project, and what sort of timeline should be expected.

There are several tools businesses can use that are specific to the second phase. They include:

* CASE (Computer Aided Systems/Software Engineering)
* Requirements gathering
* Structured analysis

### **3. Systems Design**

The third phase describes, in detail, the necessary specifications, features and operations that will satisfy the functional requirements of the proposed system which will be in place. This is the step for end users to discuss and determine their specific business information needs for the proposed system. It"s during this phase that they will consider the essential components (hardware and/or software) structure (networking capabilities), processing and procedures for the system to accomplish its objectives.

### **4. Development**

The fourth phase is when the real work begins—in particular, when a programmer, network engineer and/or database developer are brought on to do the major work on the project. This work includes using a flow chart to ensure that the process of the system is properly organized. The development phase marks the end of the initial section of the process. Additionally, this phase signifies the start of production. The development stage is also characterized by instillation and change. Focusing on training can be a huge benefit during this phase.

### **5. Integration and Testing**

The fifth phase involves systems integration and system testing (of programs and procedures)—normally carried out by a Quality Assurance (QA) professional—to determine if the proposed design meets the initial set of business goals. Testing may be repeated, specifically to check for errors, bugs and interoperability. This testing will be performed until the end user finds it acceptable. Another part of this phase is verification and validation, both of which will help ensure the program’s successful completion.

### **6. Implementation**

The sixth phase is when the majority of the code for the program is written. Additionally, this phase involves the actual installation of the newly-developed system. This step puts the project into production by moving the data and components from the old system and placing them in the new system via a direct cutover. While this can be a risky (and complicated) move, the cutover typically happens during off-peak hours, thus minimizing the risk. Both system analysts and end-users should now see the realization of the project that has implemented changes.

### **7. Operations and Maintenance**

The seventh and final phase involves maintenance and regular required updates. This step is when end users can fine-tune the system, if they wish, to boost performance, add new capabilities or meet additional user requirement

**SDLC Models**

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry .

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model
* Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models. Several models can be combined into a hybrid methodology.

Waterfall is a traditional linear and sequential approach to SDLC. The Waterfall model moves through the defined stages of SDLC only when a stage works as intended. This model can be adequate when the project is small and there are not any uncertainties in direction. However, Waterfall has fallen out of favor for SDLC adherents, due to the need for software development to be fast and flexible.

Agile is a much more utilized SDLC approach. Agile is an SDLC model that approaches software development in incremental but rapid cycles, commonly referred to as sprints. Each sprint builds upon the last. These sprints compose the project quickly with higher flexibility, as new changes in scope and direction can be implemented in each sprint. Agile can mean less time spent in the planning phases, and a project can diverge from original specifications. Documentation is crucial, regardless of the type of SDLC model for a given application,and is usually done in parallel with the development process.

**4. Describe how DNS works as you would to a novice, from the browser (from page request to page load).**

When a user types a human-readable address in his browser, the operating system’s DNS client will check for information in a local cache. If the requested address isn’t there, it will look for a Domain Name System server in local area network (LAN). When the local DNS server receives the query and the requested domain name is found, it will return the result.  
If the name is not found, the local server will forward the query to a DNS cache server, often provided by the Internet service provider (ISP). Since the DNS server’s cache contains a temporary store of DNS records, it will be able to quickly respond to requests. These DNS cache servers are called “**not authoritative DNS servers**” as they provide request resolution based in a cached value acquired from “authoritative DNS servers.”  
An**Authoritative Root name server** maintains and provides a list of authoritative name servers for each of the top-level domains (.com, .org etc).  
An**Authoritative Top level domain name server**maintains and provides a list of authoritative **name servers**for all domains (gmail.com, wikipedia.org etc). It’s job is to query name servers to find and return the authoritative name server for the requested domain.  
Now that we’ve got a better idea how DNS works, the next post will introduce you Amazon’s Route 53 and show you how easy it can be to migrate your existing domains to it.

#### **Domain Names**

A domain name is human readable name – like amazon.com – that we type in a web browser URL field. The Internet Corporation for Assigned Names and Numbers ([ICANN](https://www.icann.org/)) manages  these domain names

#### **Top Level Domain (TLD)**

TLD refers to the last part of a domain name. For example, the **.com** in amazon**.com** is the Top level domain. The most common TLDs include .com, .net, org, and .info. Country code TLDs represent specific geographic locations. For example: .in represents India. Here are some more examples:

* **com**– Commercial businesses.
* **gov**– U.S. government agencies.
* **edu**– Educational institutions such as universities.
* **org**– Organizations (mostly non-profit).
* **mil**– Military.
* **net**– Network organizations.
* **eu**– European Union.

#### **Second level Domain**

This is the part of a domain name which comes right before the TLD, “**amazon**.com”, for example.

#### **Sub Domain**

A sub domain can be created to identify unique content areas of a web site. For example, the aws of “**aws**.amazon.com”.

#### [How DNS works](https://d1o2okarmduwny.cloudfront.net/wp-content/uploads/2015/07/Screenshot-2015-07-18-23.50.46.png)**Domain Name Registrar**

By managing domain name reservations, name registrars are critical to how DNS works. ICANN [currently grants permission](https://www.icann.org/registrar-reports/accredited-list.html) to organizations to act as domain name registrars for specific higher level domains.

#### **Name Server**

Like a phone book, the name server is a collection of domain names matched to IP addresses.

### How DNS works: Domain Name System record types

**A Record**

Address records (“A Records”) map server IP addresses to domain names. For example, 72.21.206.6 to amazon.com.

**CNAME**

Canonical Name record. A CNAME record establishes one domain as an alias to another (thereby routing all traffic addressed to the alias to the target; the canonical address)

**Alias Record**

Like a CNAME record, Alias records can be used to map one address to another. But Aliases can coexist with other records using the same name.

**MX Record**

Mail Exchange Record. These records will redirect a domain’s email to the servers hosting the domain’s user accounts. Mail exchange records are used for determining the priority of email servers for a domain.

**5. Programming Challenge Description :**

You are given 3 coins of value 1, 3 and 5. You are also given a total which you have to arrive at. Find the minimum number of coins to arrive at it.

**Input:**

Your program should read lines from standard input. Each line contains a positive integer number which represents the total you have to arrive at.

**Output:**

Print out the minimum number of coins required to arrive at the total.

**PROGRAM**

function getCoinInfo(amount){

  let fiveCoins = 0;

  let threeCoins = 0;

  let oneCoins = 0;

  let pending = amount % 5;

  fiveCoins = (amount - pending)/5;

  switch(pending){

    case 4 : threeCoins = 1;

      oneCoins = 1;

      break;

    case 3 : threeCoins = 1;

      break;

    case 2 : oneCoins = 2;

      break;

    case 1 : oneCoins = 1;

  }

  console.log('five : ',fiveCoins,'threeCoins : ',threeCoins, 'oneCoins : ',oneCoins);

  console.log('total : ',fiveCoins+threeCoins+oneCoins);

}

function getCoinInfoNew(amount,coin){

  if(amount<=2)

    return amount==2?2:(amount==1)?1:0;

  else{

    let pending= amount% coin;

    coin = (amount-pending)/coin;

    return coin + getCoinInfoNew(pending,3);

  }

}

getCoinInfo(104);

console.log('From recursive :',getCoinInfoNew(104,5));

THANK YOU