1. D Why testing is required?

Ans: **Manual testing** is the process of **manually testing** software for defects. It requires a tester to play the role of an end user and use most of all features of the application to ensure correct behavior.**Testing** is **required** for an effective performance of software application or product. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It's **required** to stay in the business.

2) What types of application we test

Ans: web applications,

       desktop/windows applications

       Mobile applications

       ETL jobs

      Back end/batch programs/windows services

**ETL** stands for Extract-Transform-Load and it is a process of how data is loaded from the source system to the data warehouse. Data is extracted from an OLTP database, transformed to match the data warehouse schema and loaded into the data warehouse database

There are different stages for manual testing such as **unit testing**, **integration testing**, **system testing**, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing.

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing, by seeing the source code) tests internal structures or workings of a program,

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it

Regression testing focuses on finding defects after a major code change has occurred. Specifically, it seeks to uncover [software regressions](https://en.wikipedia.org/wiki/Software_regression), as degraded or lost features, including old bugs that have come back

A **software regression** is a [software bug](https://en.wikipedia.org/wiki/Software_bug) which makes a feature stop functioning as intended after a certain event

There are different stages for manual testing such as **unit testing**, **integration testing**, **system testing**, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing.

**Integration testing** – Testing of integrated modules to verify combined functionality after integration(client/server)

**Functional testing** – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

**End-to-end testing** – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

**Sanity testing**– Testing to determine if a new software version is performing well enough to accept it for a major testing effort

**Acceptance testing** -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

**Load testing** – Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**Ad-hoc testing**  
This type of software testing is very informal and unstructured and can be performed by any stakeholder with no reference to any test case or test design documents.  
The person performing Ad-hoc testing has a good understanding of the domain and workflows of the application to try to find defects and break the software. Ad-hoc testing is intended to find defects that were not found by existing test cases.

**API Testing**  
API testing is a type of testing that is similar to unit testing. Each of the Software APIs are tested as per API specification.

A [test plan](https://en.wikipedia.org/wiki/Test_plan) is a document detailing the objectives, target market, internal beta team, and processes for a specific beta test.

A [traceability matrix](https://en.wikipedia.org/wiki/Traceability_matrix) is a table that correlates requirements or design documents to test documents.

A [test case](https://en.wikipedia.org/wiki/Test_case) normally consists of a unique identifier, requirement references from a design specification, preconditions, events, a series of steps (also known as actions) to follow, input, output, expected result, and actual result

A [test script](https://en.wikipedia.org/wiki/Test_script) is a procedure, or programing code that replicates user actions

for a collection of test cases is a [test suite](https://en.wikipedia.org/wiki/Test_suite).

 All the test values and changeable environmental components are collected in separate files and stored as test data.

3) what is SDLC and different phases in SDLC?

Ans: Software development life cycle (SDLC) is a process to develop the application

The **systems development life cycle** (**SDLC**), also referred to as the **application development life-cycle**, is a term  to describe a process for planning, creating, testing, and deploying an information system

**4)Different phases like:**

**Requirement Analysis and planning :** Senior team members analyze the requirements/input given by customers/business users. They will check whether the requirement is feasible or not (can be done or not). They also identify the risks associated with project.

* Requirement gathering and analysis.
* Design. -In this phase the testers comes up with the [**Test strategy**](http://istqbexamcertification.com/what-are-the-test-approaches-or-strategies-in-software-testing/), where they mention what to test, how to test.
* Implementation or coding.- **Implementation / Coding:**  On receiving system design documents, the work is divided in modules/units and actual coding is started
* Testing.- [**Testing**](http://istqbexamcertification.com/what-is-a-software-testing/)**:**  After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase
* Deployment.- **Deployment:** After successful testing the product is delivered / deployed to the customer for their use.
* Maintenance. Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

Note: this high level requirements will be written in BRD (Business Requirement document) by Business Analyst

Define/Design : in the define stage Business Analyst define more details about requirements (which are in BRD) in the form of SRS (software requirement specification) or Use Case diagram.

As part of design,

Senior Developers write High Level Design Document (HLD)

Developers write Low Level Design Document (LLD)

Seniors Tester write Test Planning document

Implementation/Development: Developers write the code for the requirements

Testers write test cases as per SRS

Testing : Execute the test cases what we prepared in previous stage

Deployment : Release the tested code to production

Maintenance : Support team monitoring the system that is running in production

5) what is waterfal in SDLC?

Ans: The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

No change of requirements

6)what is the process in **agile** model

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds.

Developers start off with a simplistic project design, and then begin to work on small modules. The work on these modules is done in weekly or monthly sprints, and at the end of each sprint, project priorities are evaluated and tests are run. These sprints allow for bugs to be discovered, and customer feedback to be incorporated into the design before the next sprint is run.

No proper plan.

7)what is scrum methodology

Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Management and teams are able to get their hands around the requirements and technologies, never let go, and deliver working software, incrementally and empirically. Scrum itself is a simple framework for effective team collaboration on complex software projects.

8)what is daily standup meeting and what we discuss

A daily stand-up meeting is a short organizational meeting that is held each day. The meeting, generally limited to between five and fifteen minutes long, is sometimes referred to as a stand-up, a morning roll-call or a daily scrum.

The purpose of the meeting is for each team member to answer the following three questions:

1) What did you do yesterday?

2) What will you do today?

3) Are there any impediments in your way?

Standing, rather than sitting, reinforces the idea that the meeting is intended to be short and discourages wasted time. The stand-up is not meant to be a place to solve problems, but rather to make the team aware of current status. If discussion is needed, a longer meeting with appropriate parties can be arranged.

9)what is user story/feature/sprint back log items and tasks in user story

Epic - Create a web site (T shirt size) (L)

Story - Create the Home page (1 point)

Task - Create a button on the home page that is red (1 hour)

Features are what a system is doing. User stories are just one way amongst others to capture features.

Epic - Is simply a story, but is considered so large that it needs to be broken down into multiple stories.

Story - Is essentially a requirement that is in a low enough detail that it can be estimated.

Task - Developers, testers may break the story down further into tasks to allow them to estimate, develop and test it. (If a story has a lot of tasks, it could be worth classing it as an epic and breaking it down into multiple stories.)

10)What is sprint planning and spring retro

Sprint planning is a collaborative effort involving a ScrumMaster, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment.

Spring retro: The sprint retrospective is usually the last thing done in a sprint. Many teams will do it immediately after the sprint review. The entire team, including both the ScrumMaster and the product owner should participate.

11)what is burndown chart and velocity

Burndown Charts. The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories.

The rate of progress of a Scrum Team is called "velocity". It expresses the amount of e.g. story points completed per iteration.

12)what is product backlog item and sprint backlog items

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements.

The Sprint Backlog is a forecast (…) what functionality will be in the next Increment and the work needed to deliver that functionality. The Sprint Backlog makes visible all of the work (…) The Sprint Backlog is a plan (…)

In Scrum, a **product backlog item** ("PBI", "**backlog item**", or "**item**") is a unit of work small enough to be completed by a team in one Sprint iteration. **Backlog items** are decomposed into one or more tasks. See also **backlog** effort estimation unit

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of **product backlog items**, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

🡪>> The **sprint backlog** is a list of tasks identified by the Scrum team to be completed during the Scrum **sprint**. During the **sprint** planning meeting, the team selects some number of **product backlog** items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

13)what is user acceptance criteria test cases

User acceptance is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to Market or Production environment.

14)what is v model?

The V - model is SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model. V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage.

15)what is STLC?

Software Testing Life Cycle (STLC) is the testing process which is executed in systematic and planned manner. In STLC process, different activities are carried out to improve the quality of the product.

16)what is defect?

A defect is an error or a bug, in the application which is created. A programmer while designing and building the software can make mistakes or error. These mistakes or errors mean that there are flaws in the software. These are called defects.

17)how to arise a defect and what we specify while logging defect?

Filing defects is a very integral [part of the Software Testing Life Cycle](http://www.softwaretestinghelp.com/what-is-software-testing-life-cycle-stlc/). One of the important aspects of the defect life cycle on a generic level includes defect tracking.

Defect logging, a process of finding defects in the application under test or product by testing or recording feedback from customers and making new versions of the product that fix the defects or the clients feedback.

The number of defects gets multiplied over a period of time and to effectively manage them, defect tracking system is used to make the job easier.

defect lifecycle

Defect life cycle is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it’s not reproduced. Defect life cycle is related to the bug found during testing.

**18)Different types of testing:**

Software Testing Types:

**Black box testing** – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

**White box testing** – This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit testing** – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

**Incremental integration testing** – Bottom up approach for testing i.e continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. done by programmers or by testers.

**Integration testing** – Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

**Functional testing** – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

**System testing** – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

**End-to-end testing** – Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

**Sanity testing** – Testing to determine if a new software version is performing well enough to accept it for a major testing effort. If application is crashing for initial use then system is not stable enough for further testing and build or application is assigned to fix.

**Regression testing** – Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

**Acceptance testing** -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

**Load testing** – Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

The purpose of regression testing is to confirm that a  recent program or code change has not adversely affected existing features.

19)when do we use regression testing?

* Regression Testing is required when there is a
* Change in requirements and code is modified according to the requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix

20)when do we use integration testing?

In Integration Testing, individual software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers. Integration testing focuses on checking data communication amongst these modules.

21)Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing'.

* Interfaces of the software modules with the database could be erroneous
* External Hardware interfaces, if any, could be erroneous
* Inadequate exception handling could cause issues.

|  |  |
| --- | --- |
| Unit testing mainly focus on the testing the functionality of individual units only and does not uncover the issues arises when different modules are interacting with each other. | Integration testing is to be carried out to discover the the issues arise when different modules are interacting with each other to build overall system. |
| The goal of Unit testing is to test the each unit separately and ensure that each unit is working as expected. | The goal of Integration testing is to test the combined modules together and ensure that every combined modules are working as expected. |

22)when do we use smoke testing and sanity testing?

Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine

For Example a typical smoke test would be - Verify that the application launches successfully, Check that the GUI is responsive ... etc.

Sanity Testing is done to check the new functionality / bugs have been fixed.

The objective of the testing is to verify the "rationality" of the system in order to proceed with more rigorous testing

Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public. The focus of this testing is to simulate real users by using blackbox and whitebox techniques. The aim is to carry out the tasks that a typical user might perform.

Beta Testing of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external user acceptance testing.

23)when do we use white box testing and block box testing?

White Box Testing is the testing of a software solution's internal coding and infrastructure.It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear box testing, open box testing, logic driven testing or path driven testing or structural testing and glass box testing.

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of

Internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

23)when do we use automation testing?

* In order to break software more quickly or more effectively in a repeatable manner.
* In order to help me execute complex test cases in a more timely manner.
* In order to speed up testing.
* In order to help me set up or tear down test data (in which case my test case may or may not be automated).
* In order to improve programming skills

-----🡪 **Ad hoc testing** is an informal and improvisational approach to assessing the viability of a product. An **ad hoc test** is usually only conducted once unless a defect is found. Commonly used in software development, **ad hoc testing** is performed without a plan of action and any actions taken are not typically documented.

24)what tester will do in each phase of SDLC?

The Role of a Tester in SDLC

1. Creation of test designs, test processes, test cases and test data.
2. Carry out testing as per the defined procedures.
3. Participate in walkthroughs of testing procedures.
4. Prepare all reports related to software testing carried out.
5. Ensure that all tested related work is carried out as per the defined standards and procedures.

1. Tester prepares the Test cases, Test Scenarios from the SRS(**Software Requirements Specification)**

2. Using the script the tester performs different kinds of testing (Regression, Function)

3. Tester Notes the results(pass/Fail)

4. If Result=Fail then the scenario is raised in the Test director

5. Once its fixed by the developer the tester performs a regression testing

25)difference between load and performance testing?

Load test: any test that involves to put a determined load on an application to verify how it behaves (i.e.: response time);

Performance test: it is a load test limited by the load defined by the especification of the application - the test is to verify or confirm that the application will work at the planned performance;

Stress test: there are many meanings for stress, but I tried to educate my customers to use the term stress only when you want to load the application beyond the especification definition, to understand the maximum capacity of application before it breaks. It helps for example to foresee any problems in a near future and prepare IT teams to be aware of the application capacity;

And there is Stability test: execute a load test during a long period of time to understand how stable is an application in a long run. It helps to determine, for example, if an application has memory leaks, which is very difficult to find in a normal load test.

26)different types of non-functional testing types?

Examples of non-functional tests include:

* Load/Performance testing.
* Compatibility testing.
* Localization testing.
* Security testing.
* Reliability testing.
* Stress testing.
* Usability testing.
* Compliance testing.

27)what is test case?

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

A **test case** is a set of conditions or variables under which a tester will determine whether a system under **test** satisfies requirements or works correctly.

A **test plan** is a document detailing the objectives, target market, internal beta team, and processes for a specific beta **test** for a software or hardware product. The **plan** typically contains a detailed understanding of the eventual workflow.

A **test script** in software **testing** is a set of instructions that will be performed on the system under **test** to **test** that the system functions as expected. There are various means for executing **test scripts**

28)what is test plan/test strategy document

A test plan for software project can be defined as a document that defines the scope, objective, approach and emphasis on a software testing effort.

Ans: Test plan document contains different section like

       Types of testing :

       Exit and Entry criteria :

29)what is TDD and BDD (cucumber framework)

Its also called test-driven design, is a method of software development in which unit testing is repeatedly done on source code. Write your tests watch it fails and then refactor it. The concept is we write these tests to check if the code we wrote works fine. After each test, refactoring is done and then the same or a similar test is performed again. The process is iterated as many times as necessary until each unit is functionally working as expected. TDD was introduced first by XP. I believe I have explained enough in simple terms.

Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design

DDD-Domain Driven Testing

BDD is similar in many ways to TDD except that the word “test” is replaced with the word “Behaviour”. It’s purpose is to help the the folks devising the system (i.e., the developer) identify appropriate tests to write–that is, tests that reflect the behavior desired by the stakeholders. BDD is usually done in very English-like language helps the Domain experts to understand the implementation rather than exposing the code level tests. Its defined in a GWT format, GIVEN WHEN & THEN.

BDD focuses on:

* Where to start in the process
* What to test and what not to test
* How much to test in one go
* What to call the tests
* How to understand why a test fails

The major difference that we get to see here are

* Tests are written in plain descriptive English type grammar
* Tests are explained as behavior of application and are more user focused
* Using examples to clarify requirements

30)what is priority and severity in defect?

In software testing, defect severity can be defined as the degree of impact a defect has on the development or operation of a component application being tested.

Higher effect on the system functionality will lead to the assignment of higher severity to the bug. Quality Assurance engineer usually determines the severity level of defect

**Severity** of a defect is related to how severe a bug is. Usually the **severity** is defined in terms of financial loss, damage to environment, company's reputation and loss of life. **Priority** of a defect is related to how quickly a bug should be fixed and deployed to live servers

31)Defect Priority states the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.

Defects that leave the software system unusable are given higher priority over defects that cause a small functionality of the software to fail.

32)how to estimate test cases?

* 3-Point Software Testing Estimation Technique.- estimatedmethod is based on statistical data
* Use – Case Point Method:
* Work Breakdown Structure.- breaking down test project to small piecees
* Wideband Delphi technique.
* Function Point/Testing Point Analysis. – measure size n give weightage
* Percentage of development effort method.
* Percentage distribution.
* Best Guess.
* **Step1) Divide the whole project task into subtasks**

#### 2) Allocate each task to team member

#### Step 3) Estimate the effort for tasks

There are 2 techniques which you can apply to estimate the effort for tasks

* **Functional Point Method**
* **Three Point Estimation- best, worst, more likely test case**

#### Step 4) Validate the estimation

Once you create an aggregate estimate for all the tasks mentioned in the WBS, you need to forward it to the **management board**, who will **review** and **approve** it.

33)what is most challenge defect u came across?

1. If a defect is found and later is not reproducible then steps to reproduce that defect is a challenge.

2. I was testing a mobile app A, there was multiple features which could be added as a bookmark. While going through one flow, it was working fine.

But if via one flow, the bookmark was added and removed for Y feature and the bookmark section was accessed from second flow, then automatically bookmark was added for Y feature. After spending one full day, it took 17 steps to find the exact issue.

3. There was another app B having a field that had maximum character limit of 13 characters, I entered 12 characters alphabets and 13th character as an emoji since emoji takes 2 characters the app crashed. After this was fixed, I was not allowed to enter a emoji then I copied emoji from other place and pasted in tht field again the app crashed.

4. There was one more app C, where s filed had limit of 30 characters. I copied 31 characters from notepad in mobile and pasted it in the field and it accepted.

Basically, I have many such scenarios so I do not know what exactly you are looking for.

34)how to deal the production defects?

Ans: normally end user will report this issue.

       we need to talk to them (end users) and reproduce the issue with in staging environment

  Create defect in defect tool under the production release version

  developers will fix the issue

  we (QA) test the issue on production version code (stageing) and release the fix to proudction after we verify

 we have to create a defect on current **sprint/release** so that developer will add this code to the current sprint/release

35)test design techniques:

Broadly speaking there are two main categories of Test Design Techniques. They are:

* Static Techniques
* Dynamic Techniques
* These tests can be functional or non-functional, though usually functional. Test design techniques include: **Equivalence partitioning**, **Boundary Value Analysis**, Cause Effect Graphing. A software testing method in which the internal structure/design/implementation of the item being tested is known to the tester.

**he Techniques include:**

* Boundary Value Analysis (BVA)
* Equivalence Partitioning (EP)
* Decision Table Testing.
* State Transition Diagrams.
* Use Case Testing.

36)What is staging environment

A stage or staging environment is an environment for testing that exactly resembles the production environment. In other words, it's a complete but independent copy of the production environment, including the database. Staging provides a true basis for QA testing because it precisely reproduces what is in production. Staging (STAGE) is a pre-production environment, for final testing immediately prior to deploying to production.

staging environment as the environment used after development but before production. A staging environment is meant to have everything as closely replicated to the production environment as possible so that you can maximize your chances of finding any bugs before you release the software in production

37)what is development environment

A development environment is a collection of procedures and tools for developing, testing and debugging an application or program. The development environment normally has three server tiers, called development, staging and production.

The development environment (dev) is the environment in which changes to software are developed, most simply an individual developer's workstation. T

38)what is QA environment

A **QA environment** is where you test your upgrade procedure against data, hardware, and software that closely simulate the Production **environment** and where you allow intended users to test the resulting Waveset application. A Production **environment** is where the Waveset application is actually available for business use.

Quality analysis team make sure that the new code will not have any impact on the existing functionality and they test major functionalities of the system once after deploying the new code in their respective environment

Development, Test, QA, and Production Environments. ... A QA environment is where you test your upgrade procedure against data, hardware, and software that closely simulate the Production environment and where you allow intended users to test the resulting Waveset application.

39)what is production environment

The production environment is also known as *live*, particularly for servers, as it is the environment that users directly interact with.

A production environment is where the real-time staging of programs that run an organization are executed, and includes the personnel, processes, data, hardware, and software needed to perform day-to-day operations.

## Definition - What does *Production Environment* mean?

Production environment is a term used mostly by developers to describe the setting where software and other products are actually put into operation for their intended uses by end users

To read or write an Excel, Apache provide a very famous library POI. This library is capable enough to read and write both **XLS** and **XLSX** file format of excel.

To read **XLS** files an **HSSF** implementation is provided by POI library.

To read **XLSX** , **XSSF** implementation of **POI** **library** will be the choice. Let's study these implementations in detail.

……>>> how to update the data into XML file and read data from XML file

we demonstrate the use of **DOM parser** to modify an existing XML file :

1. Add a new element
2. Update existing element attribute
3. Update existing element value
4. Delete existing element

write code to add items to integer, string **array**

String a [];

a.add("kk" );

a.add("pp");

array:

String[] arr = new String[10]; // 10 is the length of array.

arr[0] = "kk";

arr[1] = "pp";

…

int[] series = {4,2};

int[] num = new int[args.length];