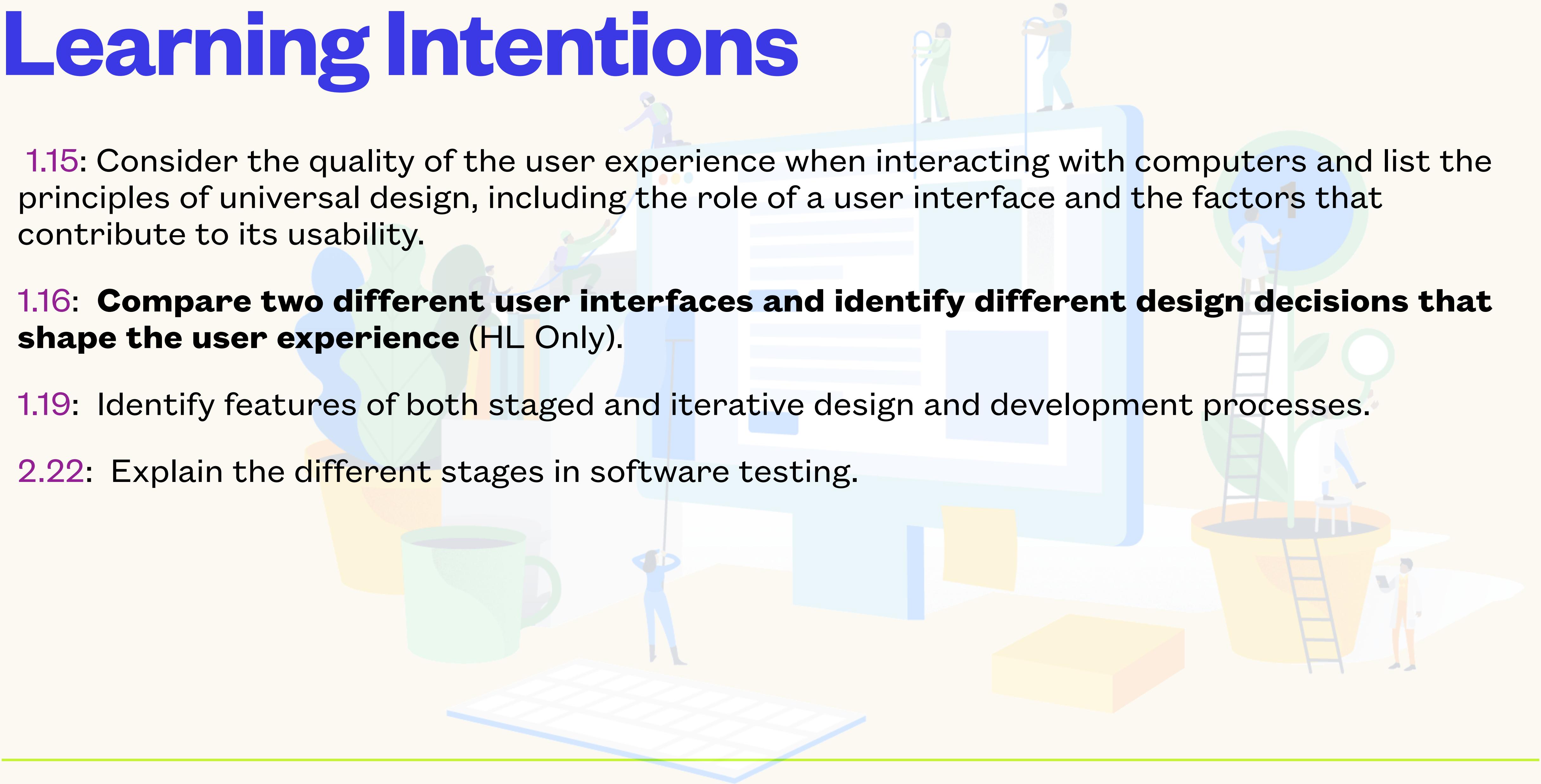




# SOFTWARE DESIGN

# Learning Intentions

- 1.15: Consider the quality of the user experience when interacting with computers and list the principles of universal design, including the role of a user interface and the factors that contribute to its usability.
- 1.16: **Compare two different user interfaces and identify different design decisions that shape the user experience (HL Only).**
- 1.19: Identify features of both staged and iterative design and development processes.
- 2.22: Explain the different stages in software testing.



# Software Development Process

- The purpose of structured software development process is to:
  1. Produce a piece of software that meets the needs of the user.
  2. Produce a piece of software on time and within budget.
  3. Minimise the cost of maintenance or upkeep of the software.

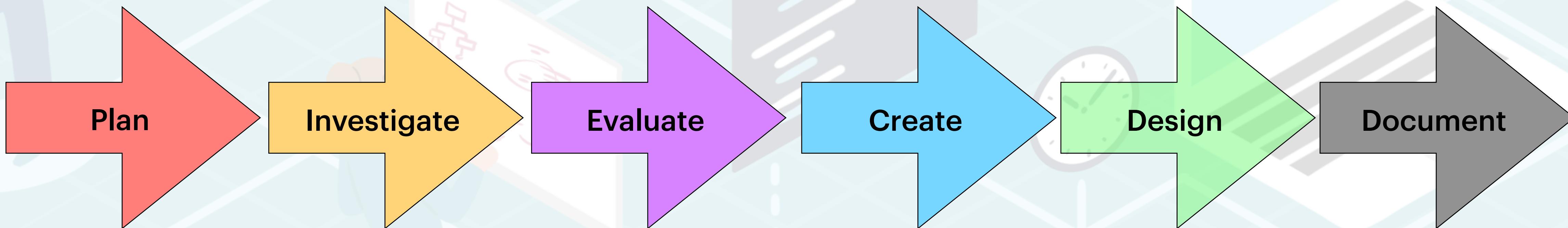
Discuss with the person next to you what you think are the steps involved in developing new software such as apps/games/programs etc?

**Make a note** of some of the steps you think will be involved in the software development process before, during and after the software is made.

# Software Development Process

Below are six steps that we will use to describe the software development process. Try to arrange them in the correct order. Is there anything missing do you think?

Hint: Think about how you developed you ALT 2, what steps did you follow?



# Iterative Design (agile)

- Iterative design is often used to describe a flexible and collaborative way of working on projects.
- Once you have identified a user need, (**Which step would this be in the development process?**), and generated a design to meet that need, you create a prototype.
- You then test/evaluate the prototype to see if it meets the user needs.
- You then take what you learn from the testing phase and amend your design and create a new prototype and begin the process all over again until you are satisfied you have reached the best possible software for release to the market.

What positives and negatives do you see with an iterative design approach?



Windows Software Iteration



iPhone Iterations



Web Archive

# Iterative Design

- Iterative design is considered an ‘Agile Approach’ to software development.

Iterative Design Positives	Iterative Design Negatives
Functional prototypes are developed early in the development life cycle.	More resources may be required. (e.g Time or Money)
Changes to project scope are easier to implement and less costly.	May be difficult to identify a strict end date for a project.
Customer feedback is based on a working prototype.	Potential to get bogged down by constant suggested improvements.
Potential problems are spotted and dealt with early.	

# Iterative Design - Key Features

- Frequent communication between developers and users.
- Flexible design process to respond to feedback from the user and understand that user requirements may change, even towards the end of a project.
- Produces a number of working prototypes.
- Keeping the design model simple, and not trying to incorporate unnecessary features.

With the person next to you try to identify some form of software you use every day, e.g an app, operating system, game, and list how you feel the user experience with that software could be improved.

Research and identify any evidence displaying how that software has already changed since its initial release, e.g through functionality changes/ interface changes

# Staged (linear) Design

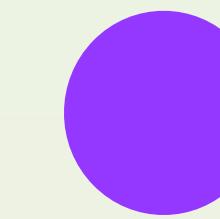
- A staged or linear development cycle is less flexible than an agile (iterative) approach.
- Each step of the development process is completed one at a time from beginning to end.
- Each step has specific outputs that lead into the next step.
- The user/customer is involved at the start of the process and in the analysis stage, otherwise they have little or no input until the evaluation stage.
- ‘Waterfall approach’ is a commonly used staged development method for software development.
- Although this is still a popular method of development it has been superseded by more effective models.

# Investigate

- This is the first step in the development cycle. The main goal of this step is to identify and analyse a problem that the software will solve.
- This stage of the development cycle begins with the gathering of information about features and requirements the final software will need.
- After the purpose and features of the software have been identified, more detailed planning is required to decide on required resources and the budget for the project.

What kind of information should be gathered or questioned asked at this stage?

# Investigate



Information Gathering Techniques

Research **three** methods developers can use to gather potential customer data to help the company identify what kind of software to develop.

1.

2.

3.

# Investigate



Fiercefun feasibility study

A feasibility study is a document produced to determine the practicality of developing the proposed software. (i.e is it a technically and economically viable solution to the existing problem).

Read the feasibility study template for FierceFun games, an Irish software developer who create games and apps for PC, console and mobile markets. While you are reading the study, do the following:

1. Identify three different features examined in a feasibility study.
2. Describe in your own words what information is covered in these three features.

# Planning

Backlog

In Progress

Peer Review

Done

A software development plan is necessary to identify the various stages of the project and to ensure that all the parties involved understand their roles and responsibilities.

- Planning is a key step in the development life cycle, it allows the involved parties to identify project plans, cost estimations, procurement requirements and dependencies.
- The planning stage should involve all relevant parties.

What kind of effects would poor/good planning have on the development process?

**3DEXPERIENCE | 3DDashboard Autonomous Mower**

Project Planner Swym Database Bookmarks Product Explorer Product Lifecycle New Tab +

Search

Petra MANN PM

ENOVIA - Project Planning - All Projects

## Autonomous Mower Project Plan

Master project plan containing all four phases of our product design cycle

State: To Do

Summary Tasks Schedule Members Content

Feb Mar Apr May Jun Jul Aug Sep

Phase 2: Prototype Design (PM)

- Drive System and Chassis Design
- Blade System Design (EE)
- Electrical & Power Systems (EE)
- Plastic Enclosure Design (PM+1)
- System Design (EE+2)
- Drawing and detailing (EE+2)

Phase 3: Analysis (PM)

- Structural Analysis (S)
- Tolerance Stack Up (EE)
- Manufacturing and Cost Analysis (MF+1)
- Thermal and Fluid Analysis (TF)
- Prototype 1 Design changes (EE+1)
- Run virtual crash test and manufacturability (TF)

Phase 4: Fabrication, Assembly, & Test (PM)

- Submit Prototype 1 for quote (MF)
- Prototype 1 fabrication (MF)
- Test Preparation (TF+1)
- Prototype 1 Assembly (EE+1)

Run virtual crash test and manufacturability

Focus on Quality

Submit Prototype 1 for quote

# Design

ADANI SMITH  
#2743



# Design

In the design stage, the desired features and operations of the software are identified.

- Processing - What algorithms and design structure will be used for the solution.
- Data structure - How will the data be stored and accessed? What type of data is available?
- Input/Output - The content to be input/output by the software and its format, medium, frequency etc.
- Security - how is the data to be kept secure from corruption or tempering?
- User Interface and Accessibility - How will the user interact with the software, will it be a GUI or Command line etc?

# Design

In the design stage, the desired features and operations of the software are identified.

- All the functionalities of the software should be identified and the algorithms that will be used to enable these functionalities .
- The flow of operations of the system can be set out using flowcharts, pseudocode, normal language or any other valid technique.
- Two important pieces of documentation are developed at this stage, a **high level design document** and a **low level design document**.

High Level Design Document	Low Level Design Document
Briefly describes each component of a piece of software	States each component's functional logic
Outlines every component's function	Features any potential error messages
Describes how the components interact and rely on each other	Contains every detail of the software interface
Features architecture diagrams	Outlines any dependency issues

# Different Design Approaches

<b>Top-Down approach</b>	<b>Modular approach</b>
Complete software system is considered as a single entity.	Software is considered as multiple different entities. I.e the UI module or the Algorithm modules (functions)
The whole system is divided into sub systems and their components.	A module is defined as a unique and addressable component which can be solved/modified without effecting other modules of the software.
Sub-systems tend to be software specific so the opportunity to reuse systems can be missed.	Similar modules from other software can be re-used.
Can be very complicated and not well structured	Very structured and easy to understand system design.

Research what a 'Prototype' is, in your research identify three different benefits to using a prototype.

# Design

## User Experience (UX)

UX Designer Jobs for Google

Career opportunities in Computer Science are endless and a lot of these roles do not involve coding. One such role is in UX Design.

- UX is known as User Experience, it refers to the experience users have when interacting with a product.
- A key component of the UX is the user interface (UI), with software this is usually a graphical interface, for example the operating system on your phone or the layout of a website.
- A user centered design approach gives priority to the needs of the user when using the software.
- At each stage of the design process, the interaction of the user and the software should be considered.
- Users themselves should be involved in this process. Why?

# Design

## User Interface (UI)

- The user interface (UI) defines how we interact with software. Interface elements consist of input controls (buttons, drop-down menus, data fields), navigational components (search fields, slider, icons, tags), informational components (progress bars, notifications, message boxes).
- A good UI focuses on making user's interactions simple and efficient.
- Types of UI include:

Graphical User Interfaces (GUI)

Command Line Interfaces (CLI)

Menu Driven Interfaces

Form Based Interfaces

Find an example of a web page **and** an app that has a bad UI and a good UI. Identify what makes these good and bad examples of UI.



# Design

## User Design (UD)

This is the Momo staircase in the vatican, it is one of the most photographed staircases in the world since its design in 1932. Who might have trouble using this staircase?



# Design

## User Design (UD)

- Why we need Universal Design
- Dog shelter problem
- 7 UD principles

What do you think is meant by accessibility?

- Universal design is a relatively new concept that is implemented when designing any product, it is a process of designing so that everyone can understand, access and use the product, regardless of age, size or ability.

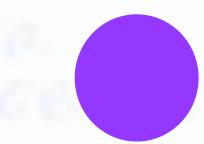
Research the seven generally agreed principles that were originally developed in 1997 by architects, product designers, engineers and environmental design researchers.

# Creation

The creation stage involves the building and development of the software. Using the design documents created in the design stage , software developers will begin to code the entire system bit by bit.

- The creation phase can be the longest phase of the development cycle, depending on the software complexity.
- Different members of a team will work on different parts of the code, these different parts can be different modules that have been identified in the design stage.
- The coding and development process will need to be documented so that the team can explain how complex software was built and why each decision was made.
- During the creation phase the code will regularly be tested or evaluated to help identify bugs, incorrect code, UX decisions etc.

# Evaluation / Testing



The software must undergo vigorous testing to ensure it works consistently as expected.

- Software must undergo numerous different types of testing during the development cycle.
- During testing, test engineers will use input data that will produce a known output, this input data can include normal, incorrect, extreme and boundary data points.

```
1 def celsius_to_kelvin(celsius_value):  
2     kelvin = celsius_value + 273.15  
3     if kelvin < 0:  
4         print(f'{celsius_value} is an invalid input')  
5     else:  
6         return kelvin  
7  
8 temp_to_convert=float(input('Please enter temperature in Celsius: '))  
9 print('Your converted temperature is',celsius_to_kelvin(temp_to_convert), '°K')
```

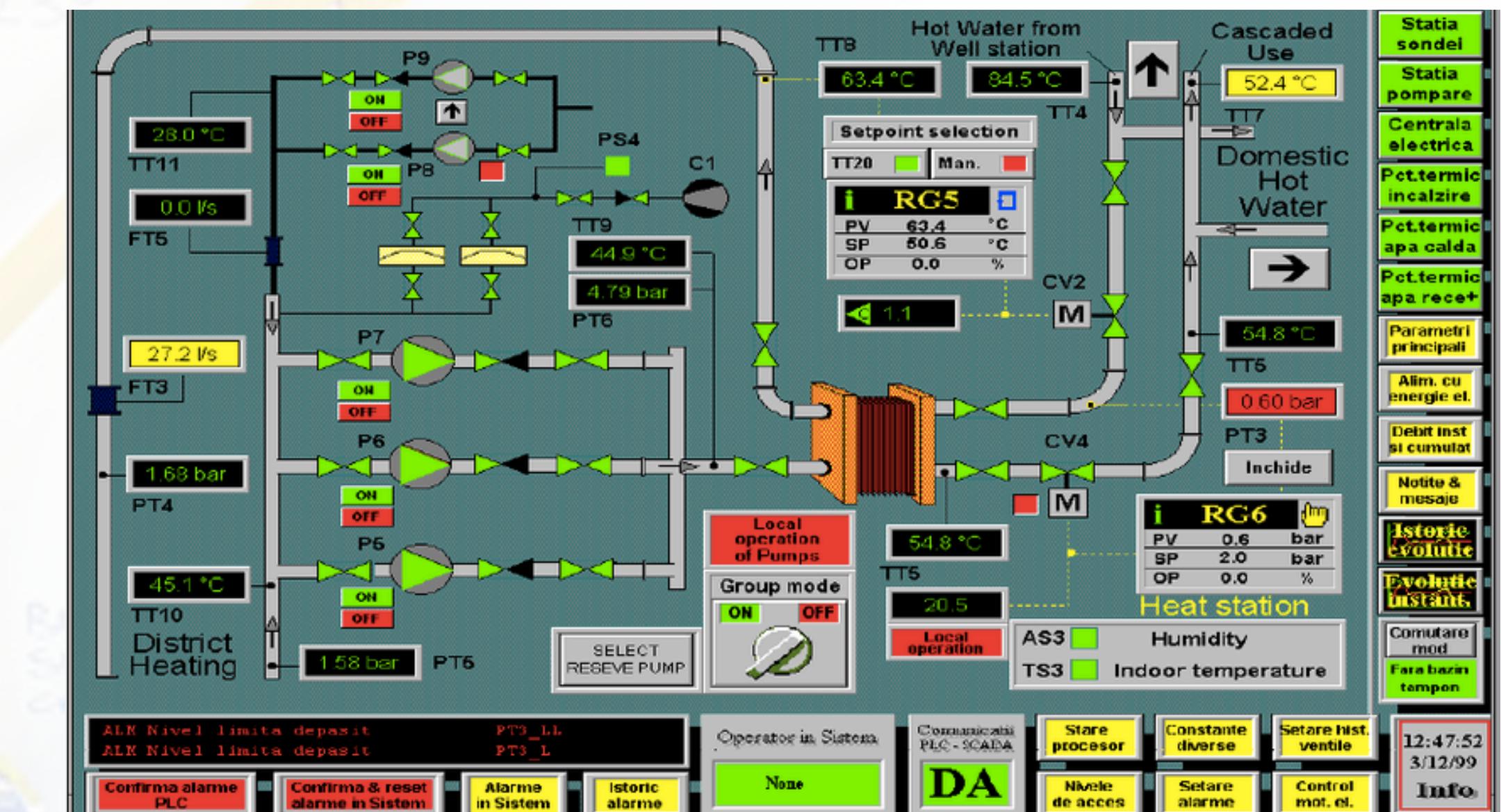
What example input data would you use to test the functionality of the above code.



# Evaluation / Testing

## Functional testing and System testing

- Functional testing involves the testing of each component (function or feature) of the software.
- Functional testing is focused on verifying the output of an action on the system.
- After identifying the function or feature to be tested, the tester will:
  - Create test data and corresponding output
  - Carry out the test
  - Compare actual output with the predicted output
  - Document the results of all tests.



# Evaluation / Testing

## Functional testing and System testing

- Testing jobs
- Vanguard Alpha

- System testing differs from functional testing as it tests the system as a whole rather than the individual components.
- System testing ensures that the software meets the agreed specification .
- System testing can involve alpha and beta testing.

Alpha Testing	Beta Testing
In-house' software testing	Software made available to different users who agree to report any problems.
Reveals errors and omissions in the system requirements definition	Product is exposed to real use. Problems/errors not identified in Alpha testing will be identified here
Tested by a dedicated testing team before release to public as Beta	Beta testing can go through different iterations

# Evaluation

## Functional testing and System testing

- Evaluation may include a post-implementation review, a critical review of the system after operation.
- Waiting period allows for users to get familiar with the software and understand the effectiveness, usability and maintainability of the software.
- An evaluation review will focus on the following:
  1. Comparison of the softwares performance against anticipated performance.
  2. Any errors that still exist or were made during the software development.
  3. Unexpected benefits.
  4. What else needs to be added or removed from the system.

# Team members

- Software development is a team based job.
- Different members of a team will have different skills and abilities.
- From the plannings stage and investigation stage a project manager can identify the skills and experience required by the project team.

Research the roles and responsibilities of the following team members:

- Project manager.
- System analyst
- Designer/Developer
- Testers
- Any other role you consider important