

BJIT Up/Cross Skill Recruitment

iOS coding aptitude test

Duration: 3 hours

Marks: 50

You can only use Objective-C, Swift, JavaScript, Java, Flutter & Kotlin to solve the problems

Problem 1:

Weight: 10

You will be given an array of integers with different values, and a target value.

You will have to write a function which will return the minimum number of values that you need to make up that target value.

If that target value cannot be made up by any combination, return -1.

You may take infinite number of each value.

Examples:

Input: values = [1,2,5], target value = 11

Output: 3

Explanation: $11 = 5 + 5 + 1$

Input: values = [2], target value = 3

Output: -1

Problem 2:

Weight: 15

Implement a ToDo app having below functionalities

1. Add button action will be a popup with 2 input fields title and description. Edit of any task will take same alert to edit todo item
2. Max 6 items can be added to this ToDo list
3. Items can be deleted, edited, skipped and undone
4. There will be custom segment control with connected or dashed line. Two done tasks will be connected with green line. If undone, line is dashed
5. Done, selected and undone indication in the item of the tab bar
6. Selection on any tab/segment: That task title and description will be shown

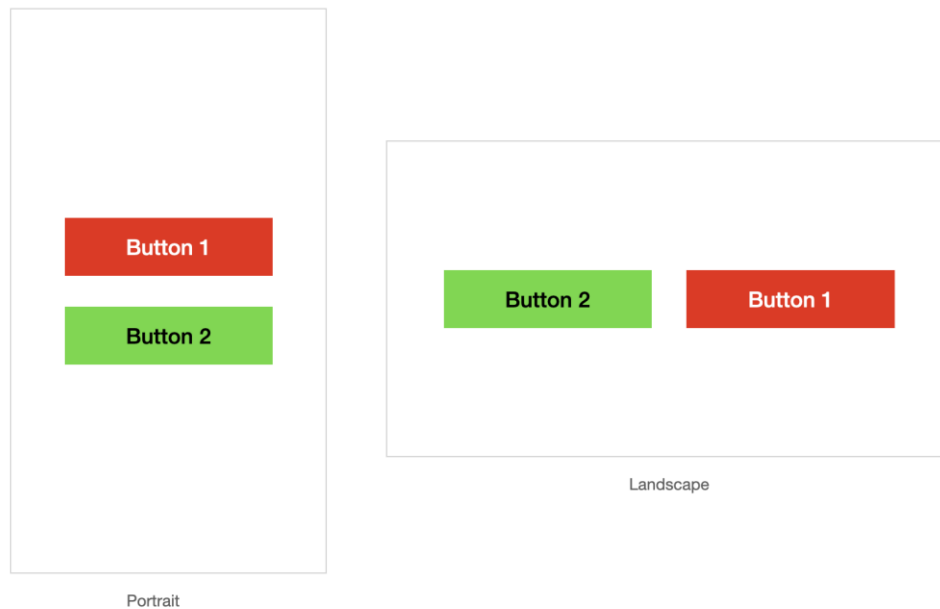
UI Design: <https://www.figma.com/file/WmdLuYVoJmMJSqe3wG3zU3/ToDo?node-id=0%3A1>

Write classes with their attributes, methods, and relationships considering the above requirements by appropriate data structure.

Note: Confirm OOP here with clean architecture(MVC or MVP, MVVM or Singleton). It is not necessary to confirm UI perfection but should confirm functional flow. No need to persist data.

Problem 3:**Weight: 10**

Create an app with one screen having 2 buttons. For portrait orientation, buttons are placed vertically & for



landscape orientation, buttons are placed horizontally as like following diagram

Note: This should have same behavior on all devices of all resolution

Problem 4:**Weight: 15**

Implement an image merger app where you need to:

- Get images from below remote locations
 - <https://i.ibb.co/VWySMMr/1.png>
 - <https://i.ibb.co/b2D0SsH/2.png>
 - <https://i.ibb.co/kHbhKX4/3.png>
 - <https://i.ibb.co/0MN8jHq/4.png>
- Merge all image into single image
- Display the merged image on UI like in the diagram

Note:

- Manage these operations with the help of multi core processor efficiently
- There is no order to get the image from remote, this are async tasks
- If one image is failed to download then use any default image from local storage given below can be used as replacement
- Confirm error handling



Merged image