**Practical – 1**

**Aim: Introduction to swift programming language, Xcode IDE and write swift program using function for following functionality:**

**a. Check whether number is prime or not**

**b. Check whether number is palindrome or not**

**Code (a):**

var num: Int = 11

if (is\_prime(x: num)) {

print("\(num) is prime")

} else {

print("\(num) is not prime")

}

func is\_prime(x: Int) -> Bool {

if x <= 1 { return false }

var counter = 0

for y in Range(1...x + 1) {

if x % y == 0 {

counter += 1

}

}

if counter == 2 { return true} else { return false }

}

**A screenshot of a computer

Description automatically generated with medium confidence**

**Code (b):**

var str = "abaaba"

if (compare(str: str)) {

print("\(str) is a palindrone")

} else {

print("\(str) is not a palindrone")

}

func compare(str: String) -> Bool {

if str == String(str.reversed()) {return true} else {return false}

}

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

**Practical – 2**

**Aim: Write a program to create parent class Person and derive two classes from it namely Student and Employee. Classes shall have following attributes and methods:**

1. **Person -> name, age, gender, city, set()**
2. **Student -> id, sem, div, sublmarks, sub2marks, sub3marks, result()**
3. **Employee -> id, designation, salary, gross\_saraly()**
4. **for gross\_salary() consider following value:**

* **If salary < 10000 then HRA=10%, DA=5%, PF=200**
* **If salary > 10000 then HRA=15%, DA=7%, PF=10%**

**Code:**

class Person {

var name: String

var age: Int

var gender: String

var city: String

init(name: String, age: Int, gender: String, city: String) {

self.name = name

self.age = age

self.gender = gender

self.city = city

}

func set() {

print(

"Name: \(name)\nAge: \(age)\nGender: \(gender)\ncity: \(city)"

)

}

}

class Student: Person {

var id: String

var sem: String

var div: Character

var sub1marks: Int

var sub2marks: Int

var sub3marks: Int

init(id: String, sem: String, div: Character,

sub1marks: Int, sub2marks: Int, sub3marks: Int) {

self.id = id

self.sem = sem

self.div = div

self.sub1marks = sub1marks

self.sub2marks = sub2marks

self.sub3marks = sub3marks

super.init(name: "Utsav Balar", age: 22, gender: "Male", city: "Surat")

}

func result() {

print(

"sub1marks: \(sub1marks)\nsub2marks: \(sub2marks)\nsub3marks: \(sub3marks)\nresult: \(sub1marks+sub2marks+sub3marks)\n"

)

}

}

class Employee: Person {

var id: String

var designation: String

var salary: Int

init(id: String, designation: String, salary: Int) {

self.id = id

self.designation = designation

self.salary = salary

super.init(name: "Utsav Balar", age: 22, gender: "Male", city: "Surat")

}

func gross\_salary() -> Double {

var HRA: Double

var DA: Double

var PF: Double

let salary = Double(self.salary)

if salary < 10000 {

HRA = 0.1 \* salary

DA = 0.05 \* salary

PF = 200

} else {

HRA = 0.15 \* salary

DA = 0.07 \* salary

PF = 0.1 \* salary

}

return salary + HRA + DA - PF

}

}

var p1 = Person(

name: "Utsav Balar",

age: 22,

gender: "Male",

city: "Surat"

)

p1.set()

var s1 = Student(

id: "201903103510391",

sem: "SEM-6",

div: "C",

sub1marks: 90,

sub2marks: 95,

sub3marks: 100

)

s1.result()

var emp1 = Employee(

id: "201903103510391",

designation: "SDE",

salary: 100000

)

print("Gross salary: \(emp1.gross\_salary())")

**Output:**

**Text

Description automatically generated**

**Practical- 3**

**Aim: Create an iOS application to develop “Say Hello App”. Use TextField to get**

**user name as input. On tap of button, display user name with hello in Label.**

**Code:**

// ViewController.swift

import UIKit

class ViewController: UIViewController {

@IBOutlet var t1: UITextField!

@IBOutlet var l1: UILabel!

@IBOutlet var l2: UILabel!

@IBAction func b1(sender: UILabel) {

l2.text = t1.text

l1.isHidden = false;

}

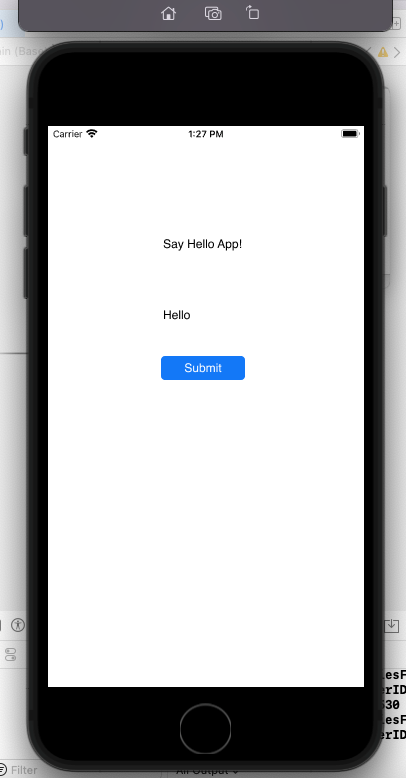
override func viewDidLoad() {

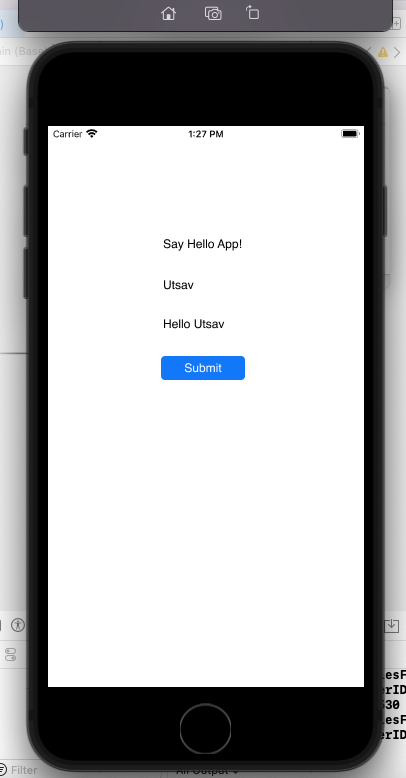
super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

}

}





**Practical - 4**

**Aim:**

**Create an iOS Application for Quiz. Create following layout given below and**

**performed following functionality.**

1. **Question and Answers Load from Data Source (Data Source contains String Array).**
2. **Contains two buttons and two labels.**
3. **Display next question on tap of Show Next Question button.**
4. **Display answer on tap of Show Answer button.**

**Code:**

import UIKit

class ViewController: UIViewController {

var questions: [String] = ["Q-1: What is the color of Orange?", "Q-2: What is the Capital of India?", "Q-3: Which is the national language of india?"]

var ans: [String] = ["orange", "new delhi", "hindi"]

@IBOutlet var tvQuestions: UITextView!

var i = 0

@IBAction func btnPrev(\_ sender: Any) {

if (i == 0) {

i = 2

} else {

i = i-1

}

tvAns.text = ""

tvQuestions.text = questions[i]

}

@IBAction func btnNext(\_ sender: Any) {

if (i == 2) {

i = 0

} else {

i = i + 1

}

tvAns.text = ""

tvQuestions.text = questions[i]

}

@IBAction func btnAns(\_ sender: Any) {

tvAns.text = ans[i]

}

@IBOutlet var tvAns: UITextView!

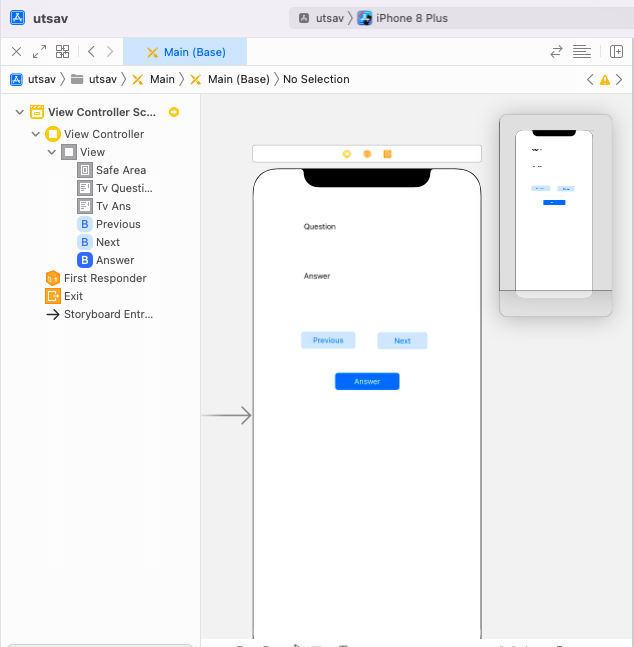
override func viewDidLoad() {

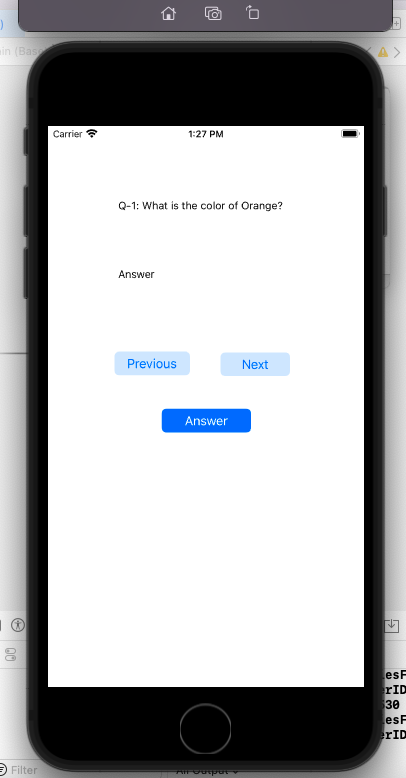
super.viewDidLoad()

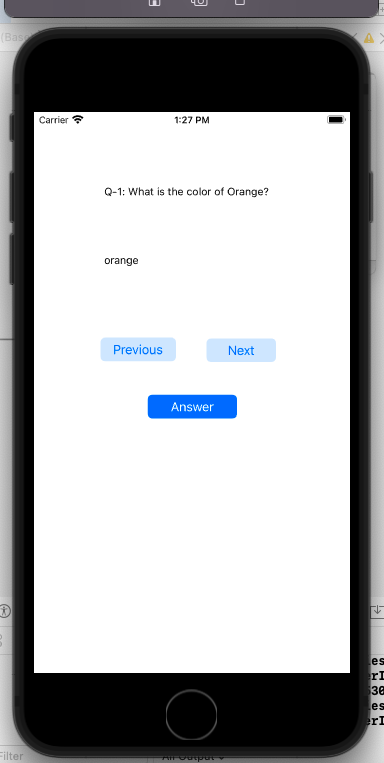
tvQuestions.text = questions[0]

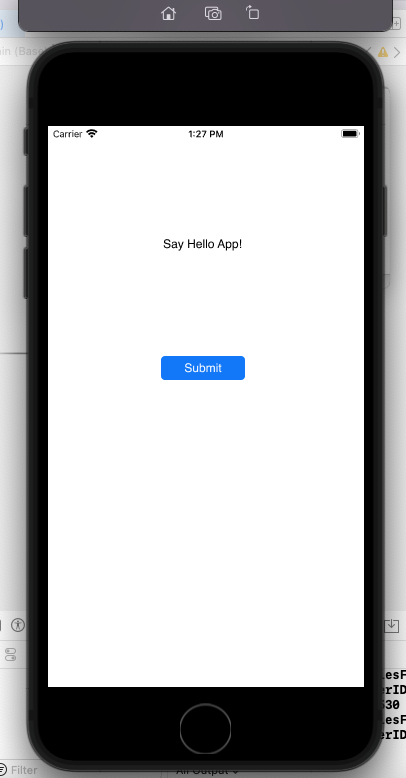
}

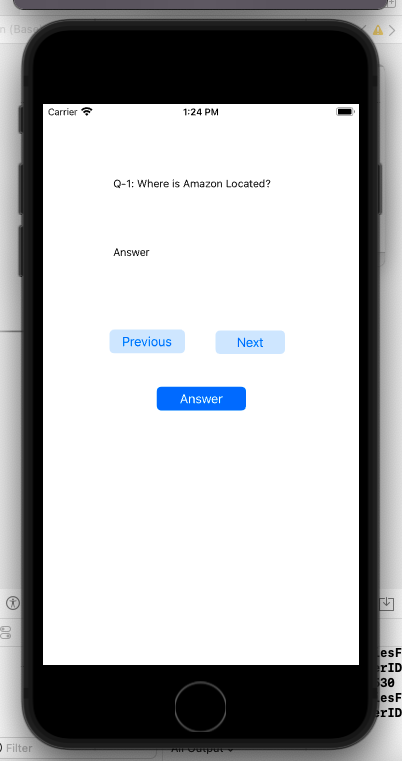
}











**Practical – 5**

**Aim: Create an IOS Application demonstrate use of tab bar control in your application.**

**Application contains following functionality:**

1. **Two tabs called Map and PrimeNumber.**
2. **On click the Map tag, Map should open.**
3. **On click of PrimeNumber, view will be displayed to check inputted Number is prime or not.**
4. **It will create 3rd tab which create another tab that checks palindrome string.**

**Code:**

**ViewController.swift**

//

// ViewController.swift

// practical5

//

// Created by bmiit on 07/03/22.

//

import UIKit

import MapKit

class ViewController: UIViewController {

@IBOutlet var prime\_input: UITextField!

@IBOutlet var prime\_output: UILabel!

@IBAction func check\_prime(\_ sender: Any) {

if let num: Int = Int(prime\_input.text!) {

if is\_prime(num: num) {

prime\_output.text = "Is a prime number"

}

else {

prime\_output.text = "Not a prime number"

}

}

}

func is\_prime(num: Int) -> Bool {

if num <= 1 {return false}

for i in 2..<num {

if num % i == 0 {return false}

}

return true

}

@IBOutlet var palindrome\_input: UITextField!

@IBOutlet var palindrome\_output: UILabel!

@IBAction func check\_palindrome(\_ sender: Any) {

let str\_t = palindrome\_input.text!

if is\_palindrome(str\_t: str\_t) {

palindrome\_output.text = "Is a Palindrome"

} else {

palindrome\_output.text = "Not a Palindrome"

}

}

func is\_palindrome(str\_t: String) -> Bool

{

if str\_t == String(str\_t.reversed()) {return true}

return false

}

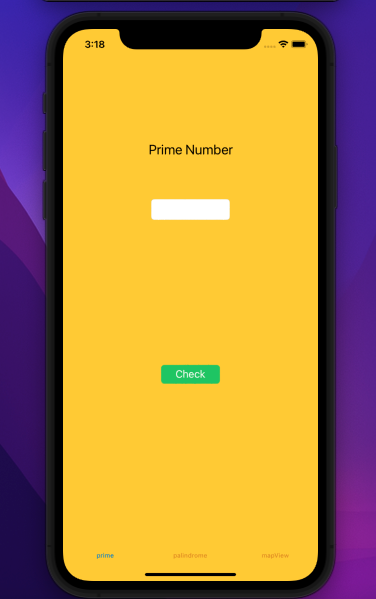
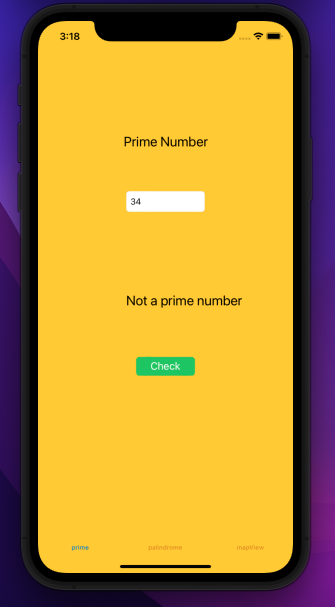
override func viewDidLoad() {

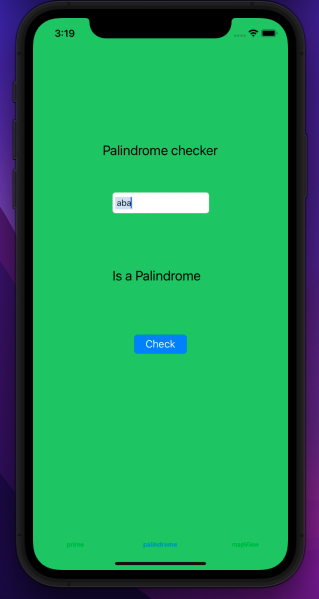
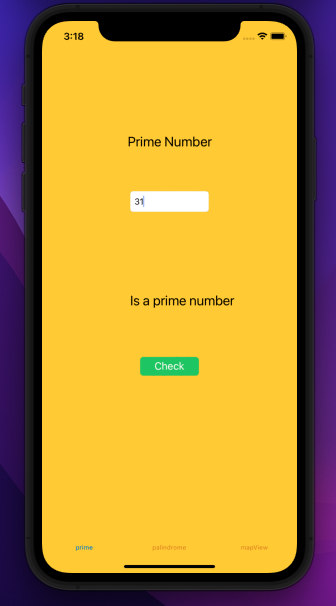
super.viewDidLoad()

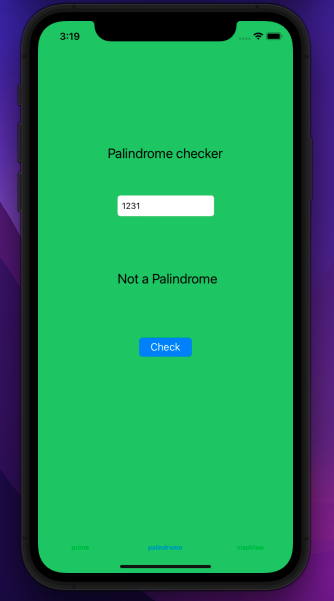
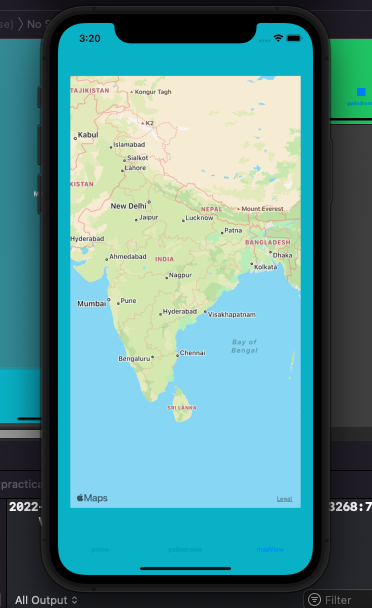
// Do any additional setup after loading the view.

}

}



**Practical: 6**

**Aim: Create an IOS interface that looks like registration view as given below.**

**When user clicks on register display the data on second view**

**Code:**

import UIKit

class View2Controller: UIViewController {

@IBOutlet var lb1: UILabel!

@IBOutlet var l2: UILabel!

@IBOutlet var l4: UILabel!

@IBOutlet var l3: UILabel!

var str1:String?

var str2:String?

var str3:String?

var str4:String?

override func viewDidLoad() {

super.viewDidLoad()

lb1.text=str1

l2.text=str2

l3.text=str3

l4.text=str4

// Do any additional setup after loading the view.

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

import UIKit

class ViewController: UIViewController {

@IBOutlet var tf1: UITextField!

@IBOutlet var tf2: UITextField!

@IBOutlet var p1: UITextField!

@IBOutlet var p2: UITextField!

@IBOutlet var tf3: UITextField!

@IBOutlet var lb3: UILabel!

@IBOutlet var d1: UIDatePicker!

@IBAction func b1(\_ sender: Any) {

if(p1.text==p2.text)

{

lb3.text = "password matched"

performSegue(withIdentifier: "con", sender: sender)

}

else

{

lb3.text = "Not matched"

}

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

if let v2=segue.destination as?

View2Controller

{

let formatter = DateFormatter()

formatter.dateStyle = DateFormatter.Style.long

formatter.timeStyle = DateFormatter.Style.short

let strDate = formatter.string(from: d1.date)

v2.str1=tf1.text! + " " + tf2.text!

v2.str2=tf3.text!

v2.str3=strDate

v2.str4=lb3.text!

}

}

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

**Output:**

