

PYTHON PROGRAMMING EXAMPLES – RASPBERRY

1.- Simple Addition Question: Write a program that takes two numbers as input and prints their sum. Solution:

```
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
sum = num1 + num2
print("The sum is:", sum)
```

2.- Circle Area Calculation Question: Write a program that takes the radius of a circle as input and calculates its area. Take the value of π from the math library. Solution:

```
import math

radius = float(input("Enter the radius of the circle: "))
area = math.pi * radius**2
print("The area of the circle is:", area)
```

3.- Vector Addition Question: Write a program that takes two 2D vectors (x and y components) as input and prints their sum. Solution:

```
x1 = float(input("Enter the x-component of the first vector: "))
y1 = float(input("Enter the y-component of the first vector: "))
x2 = float(input("Enter the x-component of the second vector: "))
y2 = float(input("Enter the y-component of the second vector: "))
sum_x = x1 + x2
sum_y = y1 + y2
print("The sum of the vectors is: ({} , {})".format(sum_x, sum_y))
```

4.- Division with Remainder Question: Write a program that takes two integers as input and calculates their quotient and remainder. Solution:

```
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
quotient = num1 // num2
remainder = num1 % num2
print("Quotient:", quotient)
print("Remainder:", remainder)
```

5.- Celsius to Fahrenheit Conversion Question: Write a program that takes a temperature in Celsius as input and converts it to Fahrenheit. Solution:

```
celsius = float(input("Enter the temperature in Celsius: "))
fahrenheit = (celsius * 9.0/5.0) + 32.0
print("Temperature in Fahrenheit:", fahrenheit)
```

6.- Even or Odd Question: Write a program that takes an integer as input and checks if it is even or odd. Solution:

```
num = int(input("Enter an integer: "))
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

7.- Circle Circumference Calculation Question: Write a program that takes the radius of a circle as input and calculates its circumference. Assume the value of π as 3.14159. Solution:

```
import math

radius = float(input("Enter the radius of the circle: "))
circumference = 2 * math.pi * radius
print("The circumference of the circle is:", circumference)
```

8.- Vector Dot Product Question: Write a program that takes two 2D vectors (x and y components) as input and calculates their dot product. Solution:

```
x1 = float(input("Enter the x-component of the first vector: "))
y1 = float(input("Enter the y-component of the first vector: "))
x2 = float(input("Enter the x-component of the second vector: "))
y2 = float(input("Enter the y-component of the second vector: "))
dot_product = x1 * x2 + y1 * y2
print("The dot product of the vectors is:", dot_product)
```

9.- Square Root Calculation Question: Write a program that takes a number as input and calculates its square root. Solution:

```
import math

num = float(input("Enter a number: "))
sqrt = math.sqrt(num)
print("The square root is:", sqrt)
```

10.- Fahrenheit to Celsius Conversion Question: Write a program that takes a temperature in Fahrenheit as input and converts it to Celsius. Solution:

```
fahrenheit = float(input("Enter the temperature in Fahrenheit: "))
celsius = (fahrenheit - 32) * 5/9
print("Temperature in Celsius:", celsius)
```

11.- Counting with a `for` Loop Question: Write a program that uses a `for` loop to print the numbers from 1 to 5. Solution:

```
for num in range(1, 6):
    print(num)
```

12.- Summing with a `while` Loop Question: Write a program that uses a `while` loop to calculate the sum of numbers from 1 to 5. Solution:

```
sum = 0
num = 1

while num <= 5:
    sum += num
    num += 1

print("Sum:", sum)
```

13.- Multiplication Table with a `for` Loop Question: Write a program that uses a `for` loop to display the multiplication table of a given number (up to 10). Solution:

```
num = int(input("Enter a number: "))

for i in range(1, 11):
    product = num * i
    print(num, "x", i, "=", product)
```

14.- Countdown with a `while` Loop Question: Write a program that uses a `while` loop to count down from 5 to 1 and prints each number. Solution:

```
num = 5

while num > 0:
    print(num)
    num -= 1
```

15.- Sum of Even Numbers with a `for` Loop Question: Write a program that uses a `for` loop to calculate the sum of even numbers from 1 to 10. Solution:

```
sum = 0

for num in range(1, 11):
    if num % 2 == 0:
        sum += num

print("Sum of even numbers:", sum)
```

16.- Factorial Calculation with a `while` Loop Question: Write a program that uses a `while` loop to calculate the factorial of a given number. Solution:

```
num = int(input("Enter a number: "))
factorial = 1
i = 1

while i <= num:
    factorial *= i
    i += 1

print("Factorial:", factorial)
```

17.- Displaying Patterns with a Nested `for` Loop Question: Write a program that uses nested `for` loops to display the following pattern:

```
*
**
***
****
*****

for i in range(1, 6):
    for j in range(i):
        print("*", end="")
    print()
```

18.- Finding Prime Numbers with a `while` Loop Question: Write a program that uses a `while` loop to find prime numbers between 1 and 10. Solution:

```
num = 1

while num <= 10:
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)
    num += 1
```

19.- Calculating Fibonacci Series with a `for` Loop Question: Write a program that uses a `for` loop to generate the Fibonacci series up to a given limit. Solution:

```
limit = int(input("Enter the limit: "))

fibonacci = [0, 1]
for i in range(2, limit):
    next_num = fibonacci[i - 1] + fibonacci[i - 2]
    if next_num <= limit:
        fibonacci.append(next_num)
    else:
        break

print("Fibonacci series:", fibonacci)
```

20.- Reversing a String with a `while` Loop Question: Write a program that uses a `while` loop to reverse a given string. Solution:

```
string = input("Enter a string: ")
reversed_string = ""

index = len(string) - 1

while index >= 0:
    reversed_string += string[index]
    index -= 1

print("Reversed string:", reversed_string)
```

11.- Controlling an LED (Digital Output) Question: Write a program to turn on an LED connected to GPIO pin 17. Solution:

```
import RPi.GPIO as GPIO
import time

# Set the GPIO mode and pin number
GPIO.setmode(GPIO.BCM)
led_pin = 17

# Set the pin as output
GPIO.setup(led_pin, GPIO.OUT)

# Turn on the LED
GPIO.output(led_pin, GPIO.HIGH)

# Wait for 5 seconds
time.sleep(5)

# Turn off the LED
GPIO.output(led_pin, GPIO.LOW)

# Cleanup GPIO
GPIO.cleanup()
```

12.- Reading a Button State (Digital Input) Question: Write a program to read the state of a button connected to GPIO pin 27 and print "Button pressed" when the button is pressed. Solution:

```
import RPi.GPIO as GPIO

# Set the GPIO mode and pin number
GPIO.setmode(GPIO.BCM)
button_pin = 27

# Set the pin as input with pull-up resistor
GPIO.setup(button_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Check button state
button_state = GPIO.input(button_pin)

if button_state == GPIO.LOW:
    print("Button pressed")

# Cleanup GPIO
GPIO.cleanup()
```

13.- Controlling a Servo Motor (PWM) Question: Write a program to control a servo motor connected to GPIO pin 18 and rotate it to different angles using PWM. Solution:

```
import RPi.GPIO as GPIO
import time

# Set the GPIO mode and pin number
GPIO.setmode(GPIO.BCM)
servo_pin = 18
```

```

# Set the pin as output
GPIO.setup(servo_pin, GPIO.OUT)

# Create PWM object with frequency and initial duty cycle
pwm = GPIO.PWM(servo_pin, 50)
pwm.start(0)

# Rotate the servo to 0 degrees
pwm.ChangeDutyCycle(2.5)
time.sleep(1)

# Rotate the servo to 90 degrees
pwm.ChangeDutyCycle(7.5)
time.sleep(1)

# Rotate the servo to 180 degrees
pwm.ChangeDutyCycle(12.5)
time.sleep(1)

# Stop PWM and cleanup GPIO
pwm.stop()
GPIO.cleanup()

```

14.- Blinking an LED with PWM Question: Write a program to blink an LED connected to GPIO pin 17 using PWM for fading effect. **Solution:**

```

import RPi.GPIO as GPIO
import time

# Set the GPIO mode and pin number
GPIO.setmode(GPIO.BCM)
led_pin = 17

# Set the pin as output
GPIO.setup(led_pin, GPIO.OUT)

# Create PWM object with frequency
pwm = GPIO.PWM(led_pin, 100)

# Start PWM with 50% duty cycle
pwm.start(50)

# Blink the LED 5 times with fading effect
for _ in range(5):
    # Increase brightness from 0% to 100%
    for duty_cycle in range(0, 101, 5):
        pwm.ChangeDutyCycle(duty_cycle)
        time.sleep(0.1)

    # Decrease brightness from 100% to 0%
    for duty_cycle in range(100, -1, -5):
        pwm.ChangeDutyCycle(duty_cycle)
        time.sleep(0.1)

# Stop PWM and cleanup GPIO
pwm.stop()
GPIO.cleanup()

```

15.- Reading Multiple Buttons (Digital Inputs) Question: Write a program to read the state of three buttons connected to GPIO pins 17, 27, and 22, and print the button numbers when pressed. Solution:

```
import RPi.GPIO as GPIO

# Set the GPIO mode and pin numbers
GPIO.setmode(GPIO.BCM)
button_pins = [17, 27, 22]

# Set the pins as inputs with pull-up resistors
for pin in button_pins:
    GPIO.setup(pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Check button states
for i, pin in enumerate(button_pins):
    if GPIO.input(pin) == GPIO.LOW:
        print("Button {} pressed".format(i+1))

# Cleanup GPIO
GPIO.cleanup()
```

16.- Generating PWM Signals with Arrays Question: Write a program to generate a PWM signal with increasing duty cycle using an array of values. Solution:

```
import RPi.GPIO as GPIO
import time

# Set the GPIO mode and pin number
GPIO.setmode(GPIO.BCM)
led_pin = 17

# Set the pin as output
GPIO.setup(led_pin, GPIO.OUT)

# Create PWM object with frequency
pwm = GPIO.PWM(led_pin, 100)

# Define duty cycle values
duty_cycles = [0, 20, 40, 60, 80, 100]

# Generate PWM signal with increasing duty cycle
for duty_cycle in duty_cycles:
    pwm.start(duty_cycle)
    time.sleep(1)

# Stop PWM and cleanup GPIO
pwm.stop()
GPIO.cleanup()
```

17.- Blinking Multiple LEDs Question: Write a program to blink three LEDs connected to GPIO pins 17, 27, and 22 simultaneously. Solution:

```
import RPi.GPIO as GPIO
import time
```

```

# Set the GPIO mode and pin numbers
GPIO.setmode(GPIO.BCM)
led_pins = [17, 27, 22]

# Set the pins as output
for pin in led_pins:
    GPIO.setup(pin, GPIO.OUT)

# Blink the LEDs simultaneously
for _ in range(5):
    GPIO.output(led_pins, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(led_pins, GPIO.LOW)
    time.sleep(0.5)

# Cleanup GPIO
GPIO.cleanup()

```

18.- Controlling LED Brightness with PWM Question: Write a program to control the brightness of an LED connected to GPIO pin 17 using PWM based on user input. Solution:

```

import RPi.GPIO as GPIO

# Set the GPIO mode and pin number
GPIO.setmode(GPIO.BCM)
led_pin = 17

# Set the pin as output
GPIO.setup(led_pin, GPIO.OUT)

# Create PWM object with frequency
pwm = GPIO.PWM(led_pin, 100)
pwm.start(0)

# Get user input for brightness level (0-100)
brightness = int(input("Enter the brightness level (0-100): "))

# Change duty cycle based on user input
pwm.ChangeDutyCycle(brightness)

# Wait for user to exit
input("Press Enter to exit...")

# Stop PWM and cleanup GPIO
pwm.stop()
GPIO.cleanup()

```

19.- Toggle an LED with a Button (Digital Input and Output) Question: Write a program to toggle an LED connected to GPIO pin 17 when a button connected to GPIO pin 27 is pressed. Solution:

```

import RPi.GPIO as GPIO

# Set the GPIO mode and pin numbers
GPIO.setmode(GPIO.BCM)
button_pin = 27
led_pin = 17

# Set the button pin as input with pull-up resistor
GPIO.setup(button_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

```



```

# Set the LED pin as output
GPIO.setup(led_pin, GPIO.OUT)

# Toggle the LED when the button is pressed
while True:
    if GPIO.input(button_pin) == GPIO.LOW:
        GPIO.output(led_pin, not GPIO.input(led_pin))
        while GPIO.input(button_pin) == GPIO.LOW:
            pass

# Cleanup GPIO
GPIO.cleanup()

```

20.- Dimming an LED with PWM and a Potentiometer Question: Write a program to control the brightness of an LED connected to GPIO pin 17 using PWM based on a potentiometer connected to GPIO pin 18. Solution:

```

import RPi.GPIO as GPIO

# Set the GPIO mode and pin numbers
GPIO.setmode(GPIO.BCM)
potentiometer_pin = 18
led_pin = 17

# Set the potentiometer pin as input
GPIO.setup(potentiometer_pin, GPIO.IN)

# Set the LED pin as output
GPIO.setup(led_pin, GPIO.OUT)

# Create PWM object with frequency
pwm = GPIO.PWM(led_pin, 100)
pwm.start(0)

# Control LED brightness based on potentiometer value
try:
    while True:
        pot_value = GPIO.input(potentiometer_pin)
        pwm.ChangeDutyCycle(pot_value)
finally:
    # Cleanup GPIO
    pwm.stop()
    GPIO.cleanup()

```