Problem 1: Traffic Light System
Problem Statement:
Write a C program to simulate a traffic light system using enum. The program should:
Define an enum named TrafficLight with the values RED, YELLOW, and GREEN.
Accept the current light color as input from the user (as an integer: 0 for RED, 1 for YELLOW, 2 for GREEN).
Display an appropriate message based on the current light:
RED: "Stop"
YELLOW: "Ready to move"
GREEN: "Go"

ANSWER:

```
#include <stdio.h>

typedef enum{
   RED=0,
   YELLOW,
   GREEN
}TrafficLight;

int main(){
   int input;
   printf("Enter the current traffic light color(0 for Red,1 For Yellow,2 For Green):");
   scanf("%d",&input);
   switch(input){
      case RED:{
         printf("\nStop\n");
         break;
      }
      case YELLOW:{
         printf("\nReady to move\n");
         break;
      }
      case GREEN:{
         printf("\nGo\n");
         break;
      }
      default:{
         printf("Invalid Inout.Please enter 0,1or 2");
      }
   }
   return 0;
}
```

Problem 2: Days of the Week
Problem Statement:
Write a C program that uses an enum to represent the days of the week. The program should:
Define an enum named Weekday with values MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, and SUNDAY.
Accept a number (1 to 7) from the user representing the day of the week.
Print the name of the day and whether it is a weekday or a weekend.
Weekends: SATURDAY and SUNDAY
Weekdays: The rest

ANSWER:

```
#include <stdio.h>
typedef enum{
    MONDAY=1,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY,
    SUNDAY
}Weekday;
int main() {
    int input;
    printf("Enter the day of the week (1 for MONDAY to 7 for SUNDAY): ");
    scanf("%d", &input);

    switch (input) {
        case MONDAY:
        case TUESDAY:
        case WEDNESDAY:
        case THURSDAY:
        case FRIDAY:
            printf("It's %s, a weekday.\n", input == MONDAY ? "Monday" :
                                            input == TUESDAY ? "Tuesday" :
                                            input == WEDNESDAY ? "Wednesday" :
                                            input == THURSDAY ? "Thursday" : "Friday");
            break;
        case SATURDAY:
            printf("It's Saturday, a weekend.\n");
            break;
        case SUNDAY:
            printf("It's Sunday, a weekend.\n");
            break;
```

```
        default:
            printf("Invalid input. Please enter a number between 1 and 7.\n");
    }
    return 0;
}
```

Problem 3: Shapes and Their Areas
Problem Statement:
Write a C program to calculate the area of a shape based on user input using enum.
The program should:
Define an enum named Shape with values CIRCLE, RECTANGLE, and TRIANGLE.
Prompt the user to select a shape (0 for CIRCLE, 1 for RECTANGLE, 2 for TRIANGLE).
Based on the selection, input the required dimensions:
For CIRCLE: Radius
For RECTANGLE: Length and breadth
For TRIANGLE: Base and height
Calculate and display the area of the selected shape.

ANSWER:

```
#include <stdio.h>

typedef enum {
    CIRCLE = 0,
    RECTANGLE,
    TRIANGLE
} Shape;

int main() {
    int input;
    printf("Select a shape (0 for CIRCLE, 1 for RECTANGLE, 2 for TRIANGLE): ");
    scanf("%d", &input);

    switch (input) {
        case CIRCLE: {
            float radius;
            printf("Enter the radius of the circle: ");
            scanf("%f", &radius);
            printf("Area of the Circle: %.2f\n", 3.14159 * radius * radius);
            break;
        }
        case RECTANGLE: {
            float length, breadth;
```

```
            printf("Enter the length and breadth of the rectangle: ");
            scanf("%f %f", &length, &breadth);
            printf("Area of the Rectangle: %.2f\n", length * breadth);
            break;
        }
        case TRIANGLE: {
            float base, height;
            printf("Enter the base and height of the triangle: ");
            scanf("%f %f", &base, &height);
            printf("Area of the Triangle: %.2f\n", 0.5 * base * height);
            break;
        }
        default:
            printf("Invalid input. Please enter 0, 1, or 2.\n");
    }
    return 0;
}
```

Problem 4: Error Codes in a Program
Problem Statement:
Write a C program to simulate error handling using enum. The program should:
Define an enum named ErrorCode with values:
SUCCESS (0)
FILE_NOT_FOUND (1)
ACCESS_DENIED (2)
OUT_OF_MEMORY (3)
UNKNOWN_ERROR (4)
Simulate a function that returns an error code based on a scenario.
Based on the returned error code, print an appropriate message to the user.

ANSWER:

```
#include <stdio.h>

typedef enum {
    SUCCESS = 0,
    FILE_NOT_FOUND,
    ACCESS_DENIED,
    OUT_OF_MEMORY,
    UNKNOWN_ERROR
} ErrorCode;

void simulateError(ErrorCode code) {
    switch (code) {
        case SUCCESS:
```

```
            printf("Operation completed successfully.\n");
            break;
        case FILE_NOT_FOUND:
            printf("Error: File not found.\n");
            break;
        case ACCESS_DENIED:
            printf("Error: Access denied.\n");
            break;
        case OUT_OF_MEMORY:
            printf("Error: Out of memory.\n");
            break;
        case UNKNOWN_ERROR:
            printf("Error: Unknown error occurred.\n");
            break;
    }
}

int main() {
    int input;
    printf("Enter an error code (0 for SUCCESS, 1 for FILE_NOT_FOUND, 2 for
ACCESS_DENIED, 3 for OUT_OF_MEMORY, 4 for UNKNOWN_ERROR): ");
    scanf("%d", &input);

    if (input >= 0 && input <= 4) {
        simulateError((ErrorCode)input);
    } else {
        printf("Invalid error code.\n");
    }
    return 0;
}
```

Problem 5: User Roles in a System
Problem Statement:
Write a C program to define user roles in a system using enum. The program should:
Define an enum named UserRole with values ADMIN, EDITOR, VIEWER, and GUEST.
Accept the user role as input (0 for ADMIN, 1 for EDITOR, etc.).
Display the permissions associated with each role:
ADMIN: "Full access to the system."
EDITOR: "Can edit content but not manage users."
VIEWER: "Can view content only."
GUEST: "Limited access, view public content only."
has context menu

ANSWER:#include <stdio.h>

```
typedef enum {
    ADMIN = 0,
    EDITOR,
    VIEWER,
    GUEST
} UserRole;

int main() {
    int input;
    printf("Enter user role (0 for ADMIN, 1 for EDITOR, 2 for VIEWER, 3 for
GUEST): ");
    scanf("%d", &input);

    switch (input) {
        case ADMIN:
            printf("Role: ADMIN\nPermissions: Full access to the system.\n");
            break;
        case EDITOR:
            printf("Role: EDITOR\nPermissions: Can edit content but not manage
users.\n");
            break;
        case VIEWER:
            printf("Role: VIEWER\nPermissions: Can view content only.\n");
            break;
        case GUEST:
            printf("Role: GUEST\nPermissions: Limited access, view public content
only.\n");
            break;
        default:
            printf("Invalid input. Please enter 0, 1, 2, or 3.\n");
    }
    return 0;
}
```