1- program defines an enumeration named week with the days of the week (Saturday to Friday). The enumeration assigns default integer values to each day, starting from zero.

sat is assigned the value 0.
sun is assigned the value 1.
mon is assigned the value 2.
tue is assigned the value 3.
wed is assigned the value 4.
thu is assigned the value 5.
fri is assigned the value 6.

يحدد البرنامج تعداداً مسمى الأسبوع مع أيام الأسبوع (من السبت إلى الجمعة). يقوم التعداد بتعيين قيم عددية افتراضية لكل يوم، بدءًا من الصفر.

تم تعيين sat القيمة 0.
يتم تعيين sun القيمة 1.
تم تعيين mon القيمة 2.
تم تعيين tue القيمة 3.
الأربعاء تم تعيين wed القيمة 4.
تم تعيين thu القيمة 5.
تم تعيين القيمة fri إلى 6.

Output

```
0
6
```

Solution

```c
//www.gammal.tech
int main() {
    enum week { sat, sun, mon, tue, wed, thu, fri };
    // Since the sequence starts naturally from zero, sat becomes 0
    printf("%d\n", sat);
    printf("%d\n", fri);
    return 0;
}
```

2- Create a program that uses an enumeration for days of the week (Monday to Sunday). Prompt the user to enter a number (1-7) and print the corresponding day.

قم بإنشاء برنامج يستخدم تعدادًا لأيام الأسبوع (من الاثنين إلى الأحد). اطلب من المستخدم إدخال رقم (1-7) وطباعة اليوم المقابل.

Input

```
Enter a number (1-7) representing a day of the week: 5
```

Output

```
Selected day: Friday
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for days of the week
enum Days {
    Monday = 1,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday,
    Sunday
};

int main() {
    // Prompt the user to enter a number representing a day
    printf("Enter a number (1-7) representing a day of the week: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Monday && userInput <= Sunday) {
        // Convert the input to the corresponding day using the enumeration
        enum Days selectedDay = userInput;

        // Print the selected day
        printf("Selected day: ");
        switch (selectedDay) {
            case Monday:
                printf("Monday");
                break;
            case Tuesday:
                printf("Tuesday");
                break;
            case Wednesday:
                printf("Wednesday");
                break;
            case Thursday:
                printf("Thursday");
                break;
            case Friday:
                printf("Friday");
                break;
            case Saturday:
                printf("Saturday");
                break;
            case Sunday:
                printf("Sunday");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 7.\n");
    }

    return 0;
}
```

3- Create a program that uses an enumeration for a simple menu with three options: Option1, Option2, and Option3. Prompt the user to enter a number (1-3) to select an option and print the selected option.

قم بإنشاء برنامج يستخدم تعدادًا لقائمة بسيطة تحتوي على ثلاثة خيارات: Option1 وOption2 وOption3. اطلب من المستخدم إدخال رقم (1-3) لتحديد خيار وطباعة الخيار المحدد.

Input

```
Enter a number (1-3) to select a menu option: 2
```

Output

```
Selected menu option: Option2
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for menu options
enum MenuOptions {
    Option1 = 1,
    Option2,
    Option3
};

int main() {
    // Prompt the user to enter a number to select a menu option
    printf("Enter a number (1-3) to select a menu option: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Option1 && userInput <= Option3) {
        // Convert the input to the corresponding menu option using the enumeration
        enum MenuOptions selectedOption = userInput;

        // Print the selected menu option
        printf("Selected menu option: ");
        switch (selectedOption) {
            case Option1:
                printf("Option1");
                break;
            case Option2:
                printf("Option2");
                break;
            case Option3:
                printf("Option3");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 3.\n");
    }

    return 0;
}
```

4- Create a program that uses an enumeration for simple directions (Up, Down, Left, and Right). Prompt the user to enter a number (1-4) to select a direction and print the selected direction.

قم بإنشاء برنامج يستخدم التعداد للاتجاهات البسيطة (Up, Down, Left, and Right). اطلب من المستخدم إدخال رقم (1-4) لتحديد الاتجاه وطباعة الاتجاه المحدد.

Input

```
Enter a number (1-4) to select a direction: 2
```

Output

```
Selected direction: Down
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for directions
enum Directions {
    Up = 1,
    Down,
    Left,
    Right
};

int main() {
    // Prompt the user to enter a number to select a direction
    printf("Enter a number (1-4) to select a direction: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Up && userInput <= Right) {
        // Convert the input to the corresponding direction using the enumeration
        enum Directions selectedDirection = userInput;

        // Print the selected direction
        printf("Selected direction: ");
        switch (selectedDirection) {
            case Up:
                printf("Up");
                break;
            case Down:
                printf("Down");
                break;
            case Left:
                printf("Left");
                break;
            case Right:
                printf("Right");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 4.\n");
    }

    return 0;
}
```

5- Create a program that uses an enumeration for simple status values (Active and Inactive). Prompt the user to enter a number (1-2) to select a status and print the selected status.

قم بإنشاء برنامج يستخدم تعدادًا لقيم الحالة البسيطة (نشط وغير نشط). اطلب من المستخدم إدخال رقم (1-2) لتحديد الحالة وطباعة الحالة المحددة.

Input

```
Enter a number (1-2) to select a status: 2
```

Output

```
Selected status: Inactive
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for status values
enum Status {
    Active = 1,
    Inactive
};

int main() {
    // Prompt the user to enter a number to select a status
    printf("Enter a number (1-2) to select a status: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Active && userInput <= Inactive) {
        // Convert the input to the corresponding status using the enumeration
        enum Status selectedStatus = userInput;

        // Print the selected status
        printf("Selected status: ");
        switch (selectedStatus) {
            case Active:
                printf("Active");
                break;
            case Inactive:
                printf("Inactive");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 2.\n");
    }

    return 0;
}
```

6- Create a program that uses an enumeration for simple weekday values (Weekday and Weekend). Prompt the user to enter a number (1-2) to select a weekday type and print the selected weekday type.

قم بإنشاء برنامج يستخدم تعدادًا لقيم أيام الأسبوع البسيطة (أيام الأسبوع وعطلات نهاية الأسبوع). اطلب من المستخدم إدخال رقم (1-2) لتحديد نوع أيام الأسبوع وطباعة نوع أيام الأسبوع المحدد.

Input

```
Enter a number (1-2) to select a weekday type: 1
```

Output

```
Selected weekday type: Weekday
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for weekday types
enum WeekdayType {
    Weekday = 1,
    Weekend
};

int main() {
    // Prompt the user to enter a number to select a weekday type
    printf("Enter a number (1-2) to select a weekday type: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Weekday && userInput <= Weekend) {
        // Convert the input to the corresponding weekday type using the enumeration
        enum WeekdayType selectedWeekdayType = userInput;

        // Print the selected weekday type
        printf("Selected weekday type: ");
        switch (selectedWeekdayType) {
            case Weekday:
                printf("Weekday");
                break;
            case Weekend:
                printf("Weekend");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 2.\n");
    }

    return 0;
}
```

7- Create a program that uses an enumeration for simple animal types (Dog, Cat, and Fish). Prompt the user to enter a number (1-3) to select an animal type and print the selected animal type.

إنشاء برنامج يستخدم تعداداً لأنواع الحيوانات البسيطة (كلب، قطة، سمكة). اطلب من المستخدم إدخال رقم (1-3) لتحديد نوع الحيوان وطباعة نوع الحيوان المحدد.

Input

```
Enter a number (1-3) to select an animal type: 3
```

Output

```
Selected animal type: Fish
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for animal types
enum AnimalType {
    Dog = 1,
    Cat,
    Fish
};

int main() {
    // Prompt the user to enter a number to select an animal type
    printf("Enter a number (1-3) to select an animal type: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Dog && userInput <= Fish) {
        // Convert the input to the corresponding animal type using the enumeration
        enum AnimalType selectedAnimalType = userInput;

        // Print the selected animal type
        printf("Selected animal type: ");
        switch (selectedAnimalType) {
            case Dog:
                printf("Dog");
                break;
            case Cat:
                printf("Cat");
                break;
            case Fish:
                printf("Fish");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 3.\n");
    }

    return 0;
}
```

8- Create a program that uses an enumeration for simple coin values (Penny, Nickel, Dime, and Quarter). Prompt the user to enter a number (1-4) to select a coin value and print the selected coin value.

قم بإنشاء برنامج يستخدم تعدادًا لقيم العملات البسيطة (Penny, Nickel, Dime, and Quarter). اطلب من المستخدم إدخال رقم (1-4) لتحديد قيمة العملة وطباعة قيمة العملة المحددة.

Input

```
Enter a number (1-4) to select a coin value: 3
```

Output

```
Selected coin value: Dime
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for coin values
enum CoinValue {
    Penny = 1,
    Nickel,
    Dime,
    Quarter
};

int main() {
    // Prompt the user to enter a number to select a coin value
    printf("Enter a number (1-4) to select a coin value: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Penny && userInput <= Quarter) {
        // Convert the input to the corresponding coin value using the enumeration
        enum CoinValue selectedCoinValue = userInput;

        // Print the selected coin value
        printf("Selected coin value: ");
        switch (selectedCoinValue) {
            case Penny:
                printf("Penny");
                break;
            case Nickel:
                printf("Nickel");
                break;
            case Dime:
                printf("Dime");
                break;
            case Quarter:
                printf("Quarter");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 4.\n");
    }

    return 0;
}
```

9- Create a program that uses an enumeration for simple traffic signal colors (Red, Yellow, and Green). Prompt the user to enter a number (1-3) to select a traffic signal color and print the selected color.

إنشاء برنامج يستخدم تعدادًا لألوان إشارات المرور البسيطة (الأحمر والأصفر والأخضر). اطلب من المستخدم إدخال رقم (1-3) لتحديد لون إشارة المرور وطباعة اللون المحدد.

Input

```
Enter a number (1-3) to select a traffic signal color: 2
```

Output

```
Selected traffic signal color: Yellow
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for traffic signal colors
enum TrafficSignalColor {
    Red = 1,
    Yellow,
    Green
};

int main() {
    // Prompt the user to enter a number to select a traffic signal color
    printf("Enter a number (1-3) to select a traffic signal color: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Red && userInput <= Green) {
        // Convert the input to the corresponding traffic signal color using the enumeration
        enum TrafficSignalColor selectedColor = userInput;

        // Print the selected traffic signal color
        printf("Selected traffic signal color: ");
        switch (selectedColor) {
            case Red:
                printf("Red");
                break;
            case Yellow:
                printf("Yellow");
                break;
            case Green:
                printf("Green");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 3.\n");
    }

    return 0;
}
```

10- Create a program that uses an enumeration for simple shapes (Circle, Square, and Triangle). Prompt the user to enter a number (1-3) to select a shape and print the selected shape.

إنشاء برنامج يستخدم تعداداً للأشكال البسيطة (دائرة، مربع، ومثلث). اطلب من المستخدم إدخال رقم (1-3) لتحديد شكل وطباعة الشكل المحدد.

Input

```
Enter a number (1-3) to select a shape: 2
```

Output

```
Selected shape: Square
```

Solution

```c
//www.gammal.tech
#include <stdio.h>

// Define the enumeration for shapes
enum Shape {
    Circle = 1,
    Square,
    Triangle
};

int main() {
    // Prompt the user to enter a number to select a shape
    printf("Enter a number (1-3) to select a shape: ");

    // Declare a variable to store user input
    int userInput;

    // Read user input
    scanf("%d", &userInput);

    // Check if the input is within the valid range
    if (userInput >= Circle && userInput <= Triangle) {
        // Convert the input to the corresponding shape using the enumeration
        enum Shape selectedShape = userInput;

        // Print the selected shape
        printf("Selected shape: ");
        switch (selectedShape) {
            case Circle:
                printf("Circle");
                break;
            case Square:
                printf("Square");
                break;
            case Triangle:
                printf("Triangle");
                break;
        }
        printf("\n");
    } else {
        // Print an error message for invalid input
        printf("Invalid input! Please enter a number between 1 and 3.\n");
    }

    return 0;
}
```