Recurring Meeting

Start a new Console application for this exercise.

In this exercise, we will simulate a simplified recurring meeting dialogue. In many calendar applications when you schedule a meeting or appointment you can chose the start date, end date, and the frequency (daily, weekly, monthly, yearly).

In this application we would like to prompt the user for the start date, end date, and frequency by letter code (D, W, M, Y) then print all of the dates per the frequency between the start and end date.

Additionally, only if the user chooses daily, we should ask the user if they want to skip weekend days, which we will define as Saturday and Sunday. If they choose Y (yes), then those dates should be omitted. For other frequencies, weekend days are always acceptable.

Here is some sample output:

```
Enter start date: 9/1/2022
Enter end date: 9/10/2022
Enter Frequency [(D)aily, (W)eekly, (M)onthly, (Y)early]: D
Skip Weekends (Y/N): Y

Meeting Dates
==============
Thursday Sep 1, 2022
Friday Sep 2, 2022
Monday Sep 5, 2022
Tuesday Sep 6, 2022
Wednesday Sep 7, 2022
Thursday Sep 8, 2022
Friday Sep 9, 2022
```

```
Enter start date: 9/1/2022
Enter end date: 9/30/2022
Enter Frequency [(D)aily, (W)eekly, (M)onthly, (Y)early]: W

Meeting Dates
==============
Thursday Sep 1, 2022
Thursday Sep 8, 2022
Thursday Sep 15, 2022
Thursday Sep 22, 2022
Thursday Sep 29, 2022
```

Every prompt should be validated and use a loop to force valid input:

- Valid dates must be entered (use TryParse)
- End Date must be after start date.
- Frequency must be one of D, W, M, or Y

- Skip weekends must be Y or N