Oregon State University
College of Engineering
ENGR 103
Project Report: Planting Schedule Generator
Alexander Vogs

Part 1: Planting Schedule Program - Design

Program Statement:
The goal of this program is to create a planting schedule for a fruit or vegetable based on when it's planted, how many days it takes to mature, and when the first frost occurs. The program should use two one-dimensional arrays to store the planting schedule (month and day), and determine how many times the crop can be planted before October 1st (frost date).

The program must:

1. Ask for the name of a crop, start month and day, and days to maturity.
2. Validate all user inputs.
3. Calculate planting dates using crop maturity + 14 days for harvest/prep.
4. Store planting dates in two arrays (months and days).
5. Print a full planting schedule.

Assumptions to be made:

1. Start month is an integer 1–12.
2. Day must be valid for that month.
3. Days to maturity is a positive integer.
4. Plants must be harvested by October 1st.
5. Each planting includes 14 days of cleanup time.

Inputs:

1. Crop name (string)
2. Start month (1–12)

3. Start day (valid for the month)

4. Days to maturity (positive int)

Outputs:

1. A planting schedule showing each valid planting date (month + day)

Sequence of Steps:

1. Ask for crop name, start month, start day, and days to maturity

2. Validate inputs:
   - Month is 1–12
   - Day is valid for that month using array month_days
   - Days to maturity > 0

3. Set frost date to October 1st

4. Use calculate_next_plant_date() to compute next date

5. Loop while the crop can still be planted before frost

6. Save month/day to arrays

7. Print schedule using print_planting_schedule()

Pseudocode:

```
Prompt for crop name
Prompt for start month
Prompt for start day
Prompt for days to maturity

Validate inputs
Initialize empty arrays for planting months and days
While next planting date is before October 1:
   Save current date to arrays
   Call calculate_next_plant_date()
Call print_planting_schedule()
```

Test Plan

Case Type	Input (Crop, Month, Day, Maturity)	Expected Output
Good Case	Tomato, 5, 10, 45	Multiple planting dates through late summer
Good Case	Lettuce, 3, 1, 30	More planting cycles possible
Edge Case	Kale, 9, 10, 15	Should only allow 1 planting
Edge Case	Cabbage, 9, 30, 2	Very short cycle, might allow one more planting
Bad Case	Carrot, 13, 5, 40	Error: Month out of range
Bad Case	Radish, 4, 31, 20	Error: April only has 30 days
Bad Case	Spinach, 5, 10, -5	Error: Days to maturity must be positive

How I'll analyze correctness:
I'll use the test plan to check valid and invalid inputs, and trace through one or two cycles by hand to make sure the date math is correct. I'll also confirm the loop stops when it should(before October 1st).

Part 2: Planting Schedule Program - Python Implementation

I wrote a Python program to generate a planting schedule for a crop using one-dimensional arrays. The program asks for the crop name, starting date, and days to maturity. It validates all inputs and uses functions to calculate the next planting date and to print the full schedule. It makes sure no planting extends beyond October 1st and includes 14 days for harvesting and prepping the soil before the next planting.

Here is the final code I used: [GitHub Download]

Revisiting / Looking Back

To test the program, I used the same cases from Part 1 and added one more crop. I picked different good, edge, and bad inputs to make sure the program handles them properly.

Test Case: Tomato Planting

Inputs:

- Crop name: Tomato
- Start month: 5
- Start day: 10
- Days to maturity: 45

Expected Output:
The schedule should allow multiple plantings from May through late summer, ending before October 1st.

Actual Output:

Planting Number	Month	Day
1	5	10
2	7	8
3	9	5

The first planting starts on May 10. After 45 days to grow and 14 days for harvest and prep, the next planting is July 8, and then again on September 5. No more cycles fit before October 1.

Test Case: Edge Case - Kale

Inputs:

- Crop name: Kale
- Start month: 9
- Start day: 10
- Days to maturity: 15

Expected Output:
Only one planting should be allowed since there's not enough time for a second one.

Actual Output:

Planting Number	Month	Day
1	9	10

Test Case: Bad Input - Invalid Month

Inputs:

- Crop name: Carrot
- Start month: 13
- Start day: 5
- Days to maturity: 40

Expected Output:
The program should reject the month and ask the user again.

Actual Output:
The program printed an error message and asked for a valid month.

How I Analyzed the Results

I used a calendar to double-check each planting cycle and made sure the day math matched the expected behavior. I confirmed that:

- All functions stayed under 20 lines
- There were no global variables
- The input validation correctly rejected invalid dates and values
- The program stopped adding plantings before the frost deadline