



## Day 10 Study Notes

### 1. Aptitude: Number System

**Goal:** Master properties of numbers to solve problems faster without full calculations.

#### Divisibility Rules

Quickly determining if a number is divisible by another is crucial for simplification.

- **2:** Last digit is even (0, 2, 4, 6, 8).
- **3:** Sum of digits is divisible by 3 (e.g., 123 → 1+2+3=6 → Divisible).
- **4:** Last two digits form a number divisible by 4 (e.g., 724).
- **5:** Last digit is 0 or 5.
- **6:** Divisible by both 2 and 3.
- **8:** Last three digits form a number divisible by 8.
- **9:** Sum of digits is divisible by 9.
- **11:** Difference between the sum of digits at odd places and even places is 0 or divisible by 11 (e.g., 1331 → (1+3) - (3+1) = 0).

#### Base Conversions

- **Decimal to Binary:** Repeatedly divide by 2 and record remainders in reverse order.
- **Binary to Decimal:** Sum of  $digit \times 2^p$ osition (positions start at 0 from right).

### 2. Programming: Prime Number Check

**Goal:** efficiently determine if a number  $N > 1$  has no divisors other than 1 and itself.

#### Logic (Optimization)

1. **Naive Approach:** Check divisibility from 2 to  $N - 1$ . ( $O(N)$ )
2. **Optimized Approach:** Check divisibility only from 2 to  $\sqrt{N}$ .
  - *Reasoning:* If  $N$  is divisible by  $p$ , then  $N = p \times q$ . If  $p > \sqrt{N}$ , then  $q$  must be  $< \sqrt{N}$ . We would have already found  $q$ .

#### Code Snippet (Python)

```
import math

def is_prime(n):
    if n <= 1:
        return False
    if n == 2: # 2 is the only even prime
        return True
```

```

if n % 2 == 0:
    return False

# Check odd numbers from 3 up to square root of n
limit = int(math.sqrt(n)) + 1
for i in range(3, limit, 2):
    if n % i == 0:
        return False

return True

```

### 3. Concept: Python Modules & Packages

**Goal:** Organize code for reusability.

#### Modules vs. Packages

- **Module:** A single file containing Python code (e.g., `calculations.py`). It can define functions, classes, and variables.
- **Package:** A directory containing multiple modules and a special file named `__init__.py` (which can be empty) that tells Python the directory is a package.

#### Importing

- `import math` → Access via `math.sqrt(25)`
- `from math import sqrt` → Access via `sqrt(25)`
- `import numpy as np` → Aliasing for brevity.

#### Benefits

1. **Namespacing:** Avoids variable name clashes (e.g., `my_module.func()` vs `other_module.func()`).
2. **Maintainability:** Breaks large codebases into manageable pieces.

### 4. C/C++ Concept: Header Files

**Goal:** Separate declarations from implementation.

#### What are they?

Header files (usually ending in `.h` or `.hpp`) contain function declarations (prototypes), macro definitions, and structure definitions. They **do not** typically contain the actual function logic (which goes in `.c` or `.cpp` files).

#### Usage

- **Standard Library:** `#include <stdio.h>` (Uses angle brackets, looks in system directories).
- **User Defined:** `#include "my_header.h"` (Uses quotes, looks in current directory first).

## Include Guards

To prevent a header file from being processed multiple times in a single compilation, use guards:

```
#ifndef MY_HEADER_H
#define MY_HEADER_H

// Declarations go here
void my_function(int a);

#endif
```

## 5. SQL: LIKE Operator

**Goal:** Filter data based on string patterns rather than exact matches.

### Wildcard Characters

- % : Represents zero, one, or multiple characters.
- \_ : Represents exactly one single character.

### Syntax

```
SELECT column1, column2
FROM table_name
WHERE columnN LIKE pattern;
```

### Common Examples

Pattern	Description	Example Match
'A%'	Starts with "A"	Apple, Ant
'%a'	Ends with "a"	Banana, Soda
'%or%	Contains "or"	Orange, World
'_r%'	Second letter is "r"	Bread, Train
'a__'	Starts with "a" and is 3 chars long	ant, art

### Example Query

Find all employees whose names start with 'J':

```
SELECT * FROM Employees
WHERE Name LIKE 'J%';
```

