

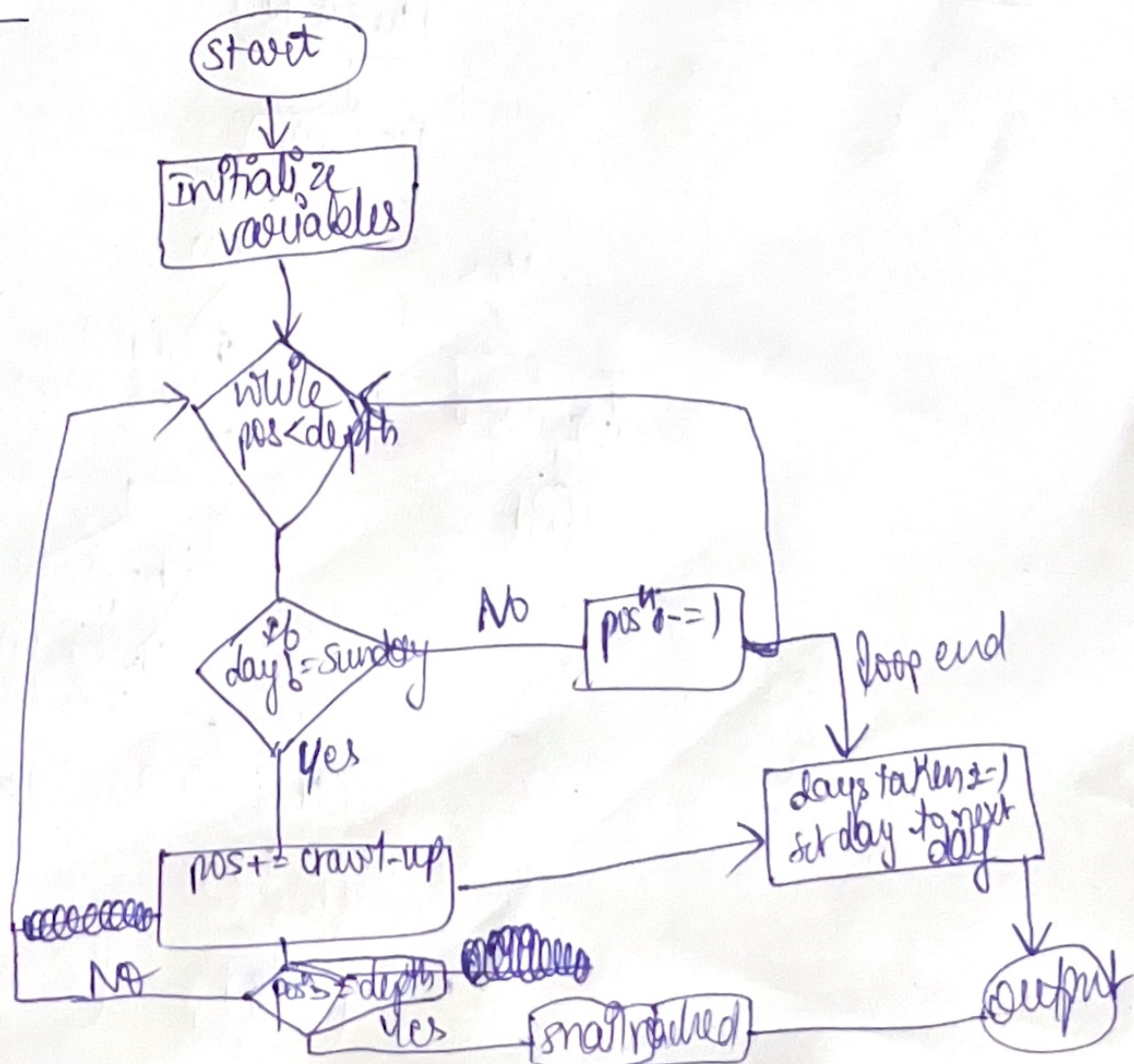
Teeranshu Goel

Problem 1) Snail in well problem

Algo:

- (1) initializing variables like depth=22, crawl-up=4, slip=2, day=Monday, position=0, days-taken=0
- (2) Looping to calculate snail's moves:
- while (position < depth)
 - if (day != Sunday)
 - then position += crawl-up
 - if position >= depth
print("snail has reached out of well")
break
 - Else
→ position -= 1 (or no. of days snail slip on Sunday)
 - days-taken += 1
 - Set day to next day in week (Monday ... Sunday → Monday)
- (3) Print days-taken

* Flowchart

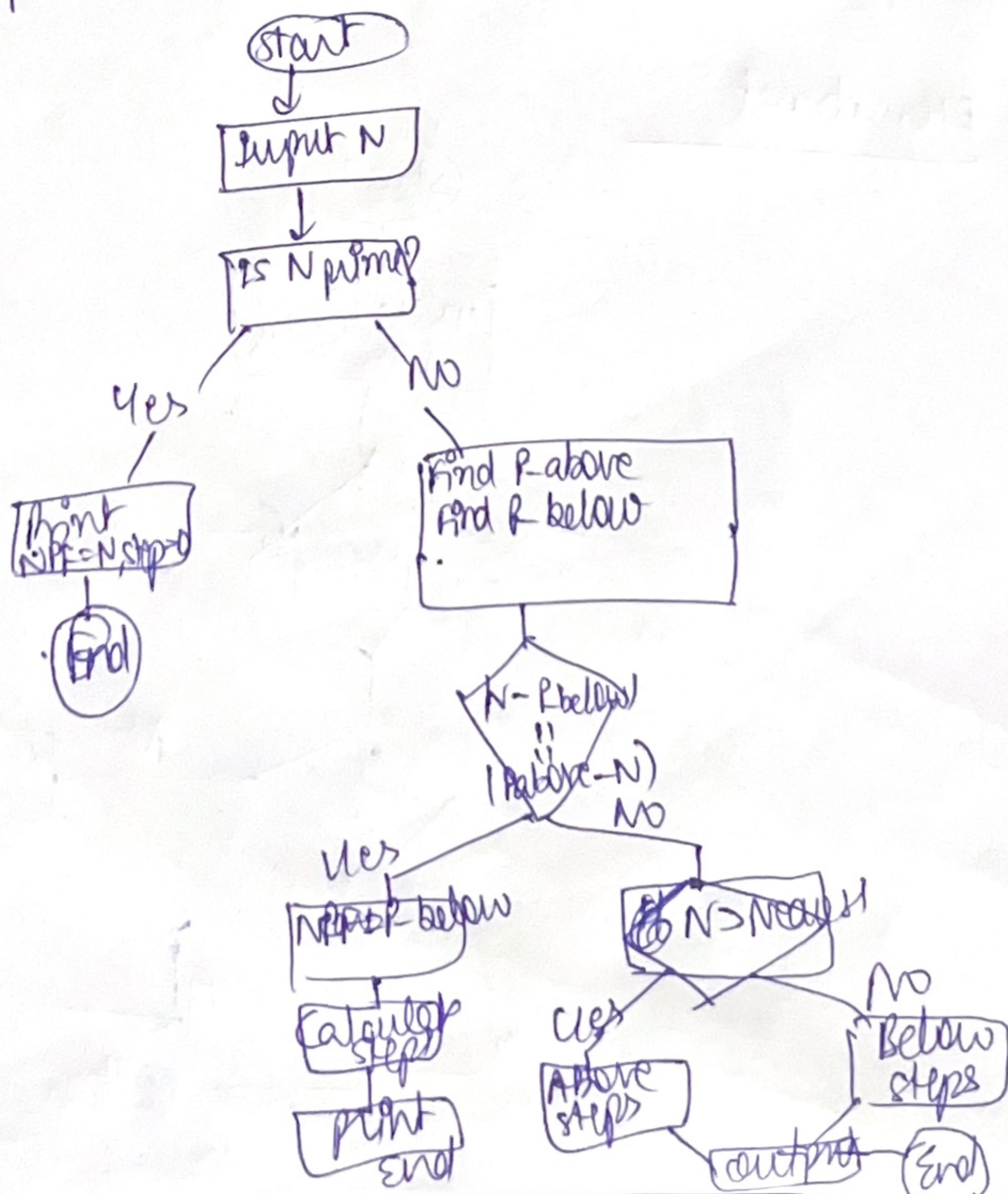


Problem 2 Building floors $\rightarrow$ Prime no. problem

Algorithm

- (1) input select floor N
- (2) check N is a prime number
If N is prime, no walking needed so, ~~output~~
output: nearest prime floor = 0, steps = 0
- (3) If N is not prime
Find nearest prime above & (P-above)
below N (P-below)
If $|N - P\text{-below}| == |P\text{-above} - N|$ (prefer P-below).
Else choose closer prime floor
- (4) Calculate steps
If above: steps = $(N - \text{nearest prime floor}) * 10$ (walk up).
If below: steps = $(\text{nearest prime floor} - N) * 10$ (walk down).
- (5) output: nearest prime floor
• steps to walk

Flowchart



Problem 3

Approach:

- (1) convert the number to string
- (2) Pad string with leading zeros for grouping of two (multiple of two)
- (3) Group the digits eg:
03, 98, 37, 857
- (4) convert group to words for units, tens, hundreds
- (5) combine words appropriately

Algo:

- (1) Input N
- (2) num-str = str(N)
- (3) Pad the string with leading zeros.
- (4) Group string into crores, lakhs, thousands etc
- (5) convert each group into words:
→ use maps
- (6) combine words eg:
3 → Three Crore
98 → Ninety-eight Lakh
- (7) output

* Flowchart:

