

Bài 2 : số nguyên mới

1.Abstraction

Ta có thể thấy được rằng đây là một bài toán ngắn gọn và dễ hiểu.

Bài tập yêu cầu chúng ta phải tạo thành 1 số nguyên mới chỉ khác với số nguyên n ban đầu duy nhất 1 chữ số và số đó phải chia hết cho 3 và số đó phải là số lớn nhất có thể .

Cách tiếp cận ,

2.Patter Recognition

- theo cách làm của nhóm thì trước tiên chúng em sẽ kiểm tra xem số đó có chia hết cho 3 hay ko

nếu chia hết cho 3 thì đơn giản chỉ việc cộng thêm 3 vào số đầu tiên cho đến khi nào số đó vượt qua 9 thì dừng lại và xuất kết quả .

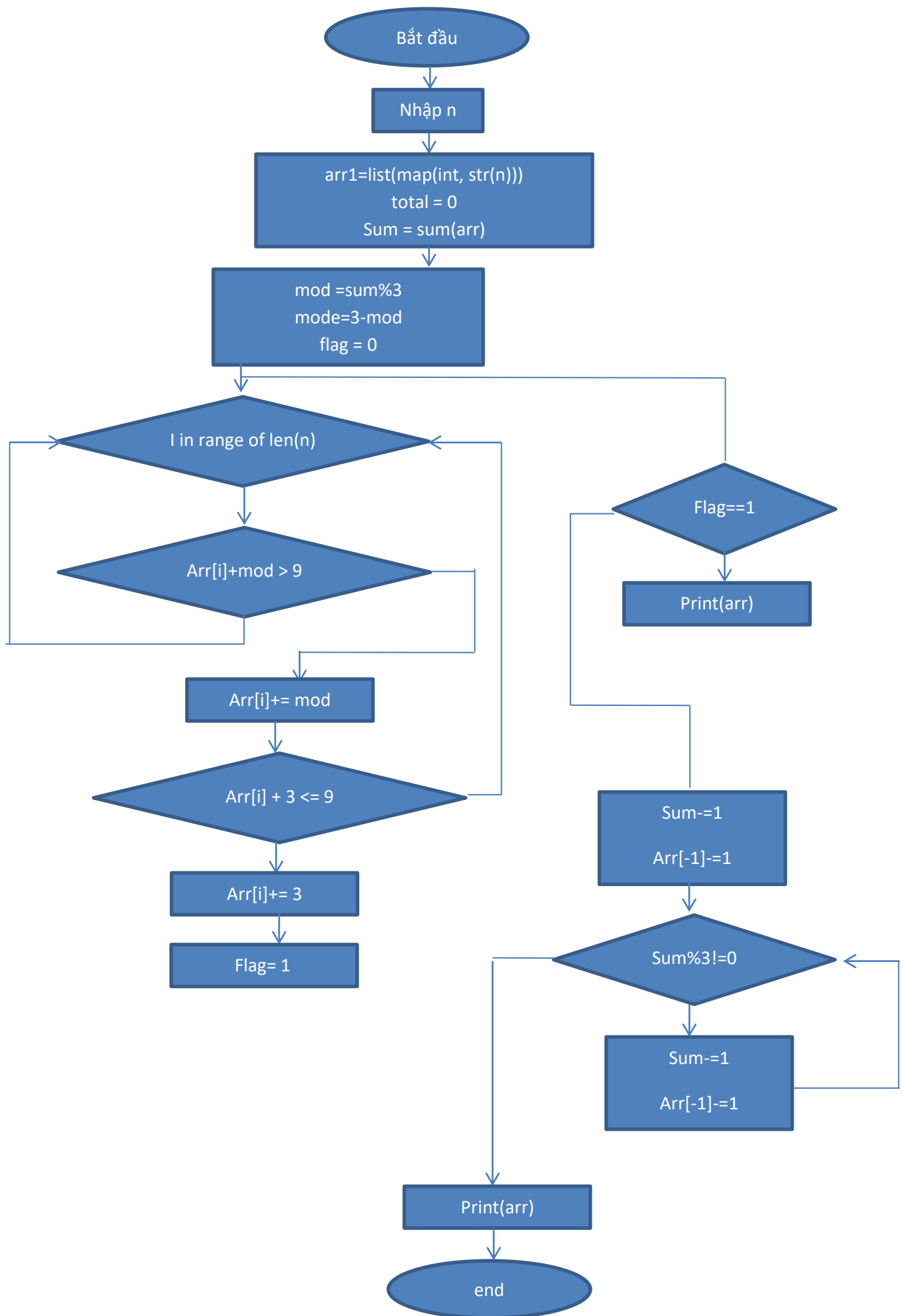
lưu ý :vì có khả năng số đầu tiên sẽ là số 9 nên t phải đặt điều kiện vòng lặp (nếu số đầu tiên + cho 3) >9 hoặc số đó == 9 thì chúng t phải lùi về 1 đơn vị và kiểm tra số đó. Nếu thỏa mãn thì chúng ta thực hiện phép +3

- nếu số đó không chia hết cho 3 thì t phải kiểm tra xem số đó chia 3 dư bao nhiêu , nếu dư 2 thì + thêm 1 , dư 1 + thêm 2 khi đó thì sẽ thành số chia hết cho 3 , làm như trên

- nếu số này quá lớn như 88 ko thể tăng đc nữa thì t phải giảm về 87 để chia hết cho 3 và max

- nếu số này là 99 thì chia hết cho 3 nhưng mà ko khác 1 số , nên t phải giảm 3 t sẽ có 96

Độ phức tạp xấu nhất sẽ là lượng chừng $O(n)$



Code

```
import math
n = int(input());
arr1=list(map(int, str(n)))
total = 0
sum = 0
for i in range(0,len(arr1)):
    sum+=arr1[i]

mod = sum%3
mod = 3 - mod

done = False

for i in range(0, len(arr1)):
    if arr1[i] + mod > 9:
        continue
    arr1[i]+=mod
    while arr1[i] + 3 <= 9:
        arr1[i] +=3
    done = True
    break

if done == True:
    print(*arr1, sep="")
else:
    sum -= 1
    arr1[-1]-=1
    while sum%3!=0:
        sum-=1
        arr1[-1]-=1
    print(*arr1, sep="")
```