Name: Yash Sarang

Div: DIAD Roll No: 47

Seat No: 4053339

Subject: C.P.

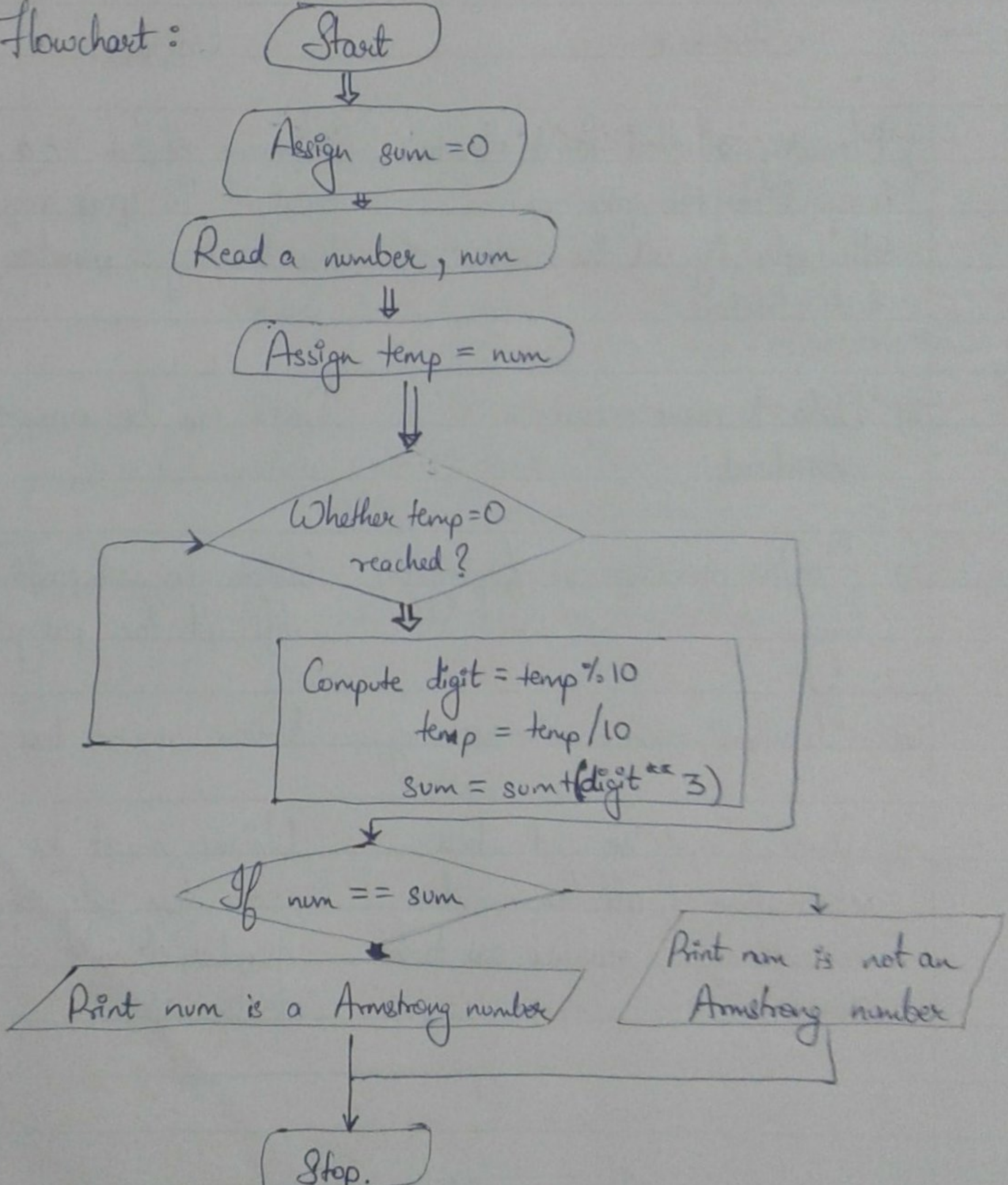
Date: 14/8/2021

Signature: Sarangyash

Page no: 1/5.

Q2.B. Program/Algorithm to check a given number is a Armstrong or not.

Flowchart :



Q2.B. Algorithm:

```
Step 1: Start
Step 2: Declare variable sum, temp, num.
Step 3: Read num from User.
Step 4: Initialize Variable sum = 0 and temp = num.
Step 5: Repeat Until num >= 0.
        5.1  sum = sum + cube of last digit.
                i.e [(num % 10) (num % 10) (num % 10)]
        5.2  num = num / 10.
Step 6: If  sum == temp
            Print "Armstrong Number".
        Else
            Print "Not an Armstrong Number".
Step 7: Stop.
```

Q2.

c)

→

	Structure	Union
i)	Memory allotted for a structure is equal to the space required collectively by all the members of that structure.	Memory allotted for a union is equal to the space required by the largest member of that union.
ii)	Data is more secure in structures	Data can be corrupted in a union.
iii)	Structure provides ease of programming	Unions are comparitively difficult for programming
iv)	Structure requires more memory	Union requires less memory
v)	Structures must be used when information of all the member elements of a structure are to be stored	Unions must be ~~stored~~ used when only one of the member elements of the union is to be stored.

Q 2 E.

→ Code:

```
# include <stdio.h>
void main ()
{
    int a=10, b=15;
    printf ("10& 15 = %d\n", a&b);
    printf (" 10|15 = %d\n",    a|b );
    printf (" 10^15 = %d \n",   a^b );
    printf ("~10 = %d\n",       ~a );
    printf (" 10 <<2 = %d\n",   a<<2);
    printf (" 10 >>2 = %d\n",   a>>2);
}
```