Name:- Udoy Saha
ID :- 23341134

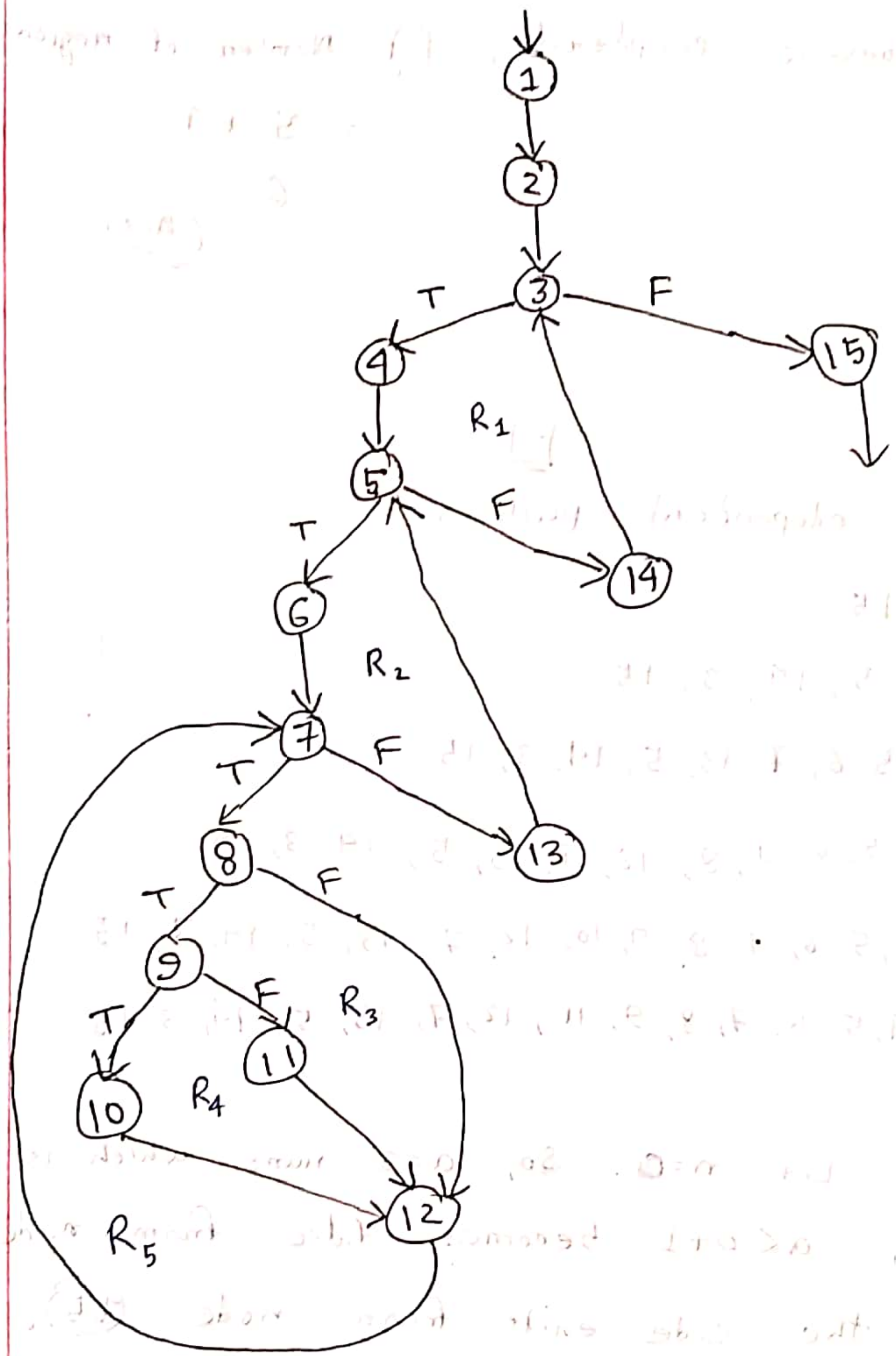
CSE 470
section :- 04

Ans to the ques no :- 1

1

```
public void find_pythagorean_triples ( int n) { —— (1)
    for ((int a=1) ; (a< n+1) ; (a++)) {
        for ((int b= a) ; (b< n+1) ; (b++)) {
            for ((int c=b) ; (c<n+1) ; (c++)) {
                if (((a*a)+(b*b) ==(c*c))) { —— (8)
                    if(a%5==0|| b%5==0 || c%5==0) { —— (9)
                        system.out.println("... by 5"); } —— (10)
                    }
                    else {
                        system.out.println("..."+c); } —— (11)
                    }
                }
            }
        }
    } —— (15)
}
```

The CFG for the code given:-



**2**

The cyclomatic complexity, M = Number of regions + 1

= 5 + 1

= 6

(Ans)

**3**

All the independent paths :

(i) 1, 2, 3, 15

(ii) 1, 2, 3, 4, 5, 14, 3, 15

(iii) 1, 2, 3, 4, 5, 6, 7, 13, 5, 14, 3, 15

(iv) 1, 2, 3, 4, 5, 6, 7, 8, 12, 7, 13, 5, 14, 3, 15

(v) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 7, 13, 5, 14, 3, 15

(vi) 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 7, 13, 5, 14, 3, 15

Test case:- Let, n = 0. So, a = 1 runs which is node 2, then, a < 0+1 becomes false from node 3. Finally the code exits from node 15.

The path becomes 1 → 2 → 3 → 15.

Indeed it is an independent path and the test is successful.

Here, number of independent paths $\leq$ Cyclomatic complexity

So, our path based testing is done correctly. Also, the path generated right output for its input.

— o — x — o —

Ans to the ques no:- 2

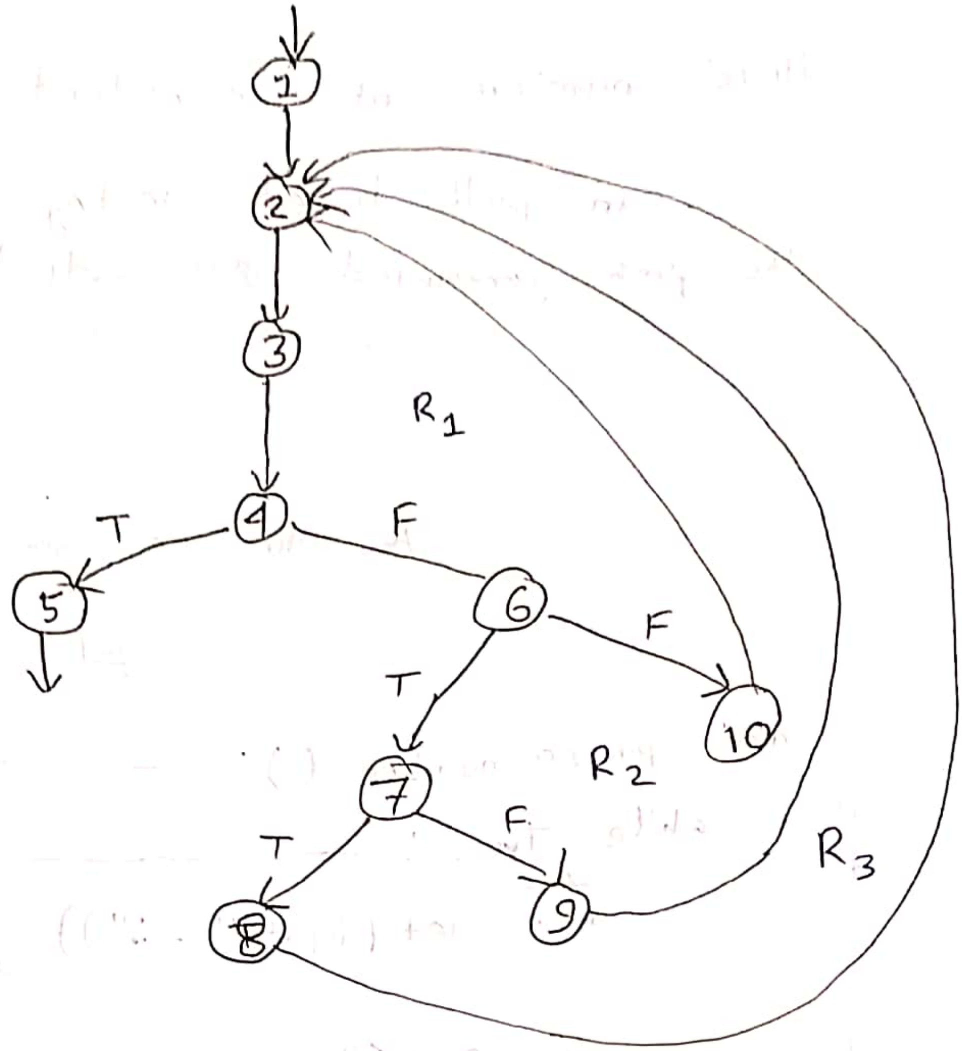
[1]

```
def process_numbers():                      —— (1)
    while True:                             —— (2)
        # . . . .
        num = int(input(". . .:"))  }       —— (3)

        if num < 0:                         —— (4)
            # . . . .
            print("... entered")    }       —— (5)
            break                   }

        if num % 2 == 0:                    —— (6)
            if num % 3 == 0:                —— (7)
                # . . . .
                print("... by 3")   }       —— (8)
            else:
                # . . .
                print(num)          }       —— (9)
        else:
            # . . . .
            print(num)              }       —— (10)
```

∴ The CFG :-




$\boxed{2}$

Here, cyclomatic complexity = Number of Regions + 1

= 3 + 1

= 4 (Ans)

3

All the independent paths:-

(i) 1, 2, 3, 4, 5

(ii) 1, 2, 3, 4, 6, 10, 2, 3, 4, 5

(iii) 1, 2, 3, 4, 6, 7, 9, 2, 3, 4, 5

(iv) 1, 2, 3, 4, 6, 7, 8, 2, 3, 4, 5

Test:- After passing node 1 and 2, let the input be -5, in node 3. Now nod 4 produces true, prints "Negative number entered", and ends the program from node 5.

So, path is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$.

Indeed its an independent path and the test case is successful.

4

Here, number of independent paths $\leq$ Cyclomatic complexity.

Thus, our path-based testing is done correctly. Also, the path generated right output.

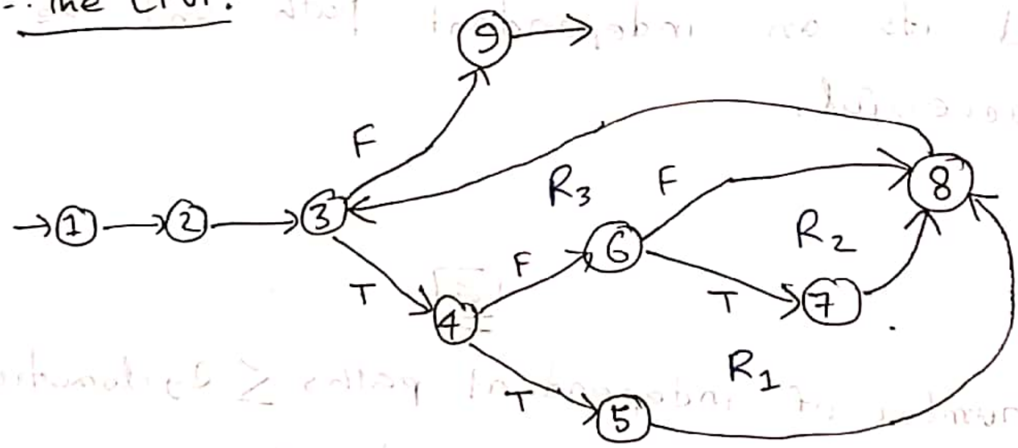
## Answer to the ques. no:- 3

$\boxed{1}$

**Code 1:-**

```
int sum=0  ------------------------------------ 1
      (2)              (3)           (8)
for ((int i=0); (i < A.length); (i++))
{
   if(A[i] % 2 == 0)  -------------------------- 4
   {    sum += A[i];}  ------------------------- 5
   }
   else
   {
      if (A[i] % 2 != 0)  ---------------------- 6
      {    System.out.println (A[i]+" is odd, skipping");} ---- 7
      }
   }
}
System.out.println(sum);  ---------------------- 9
```

**∴ The CFG :-**

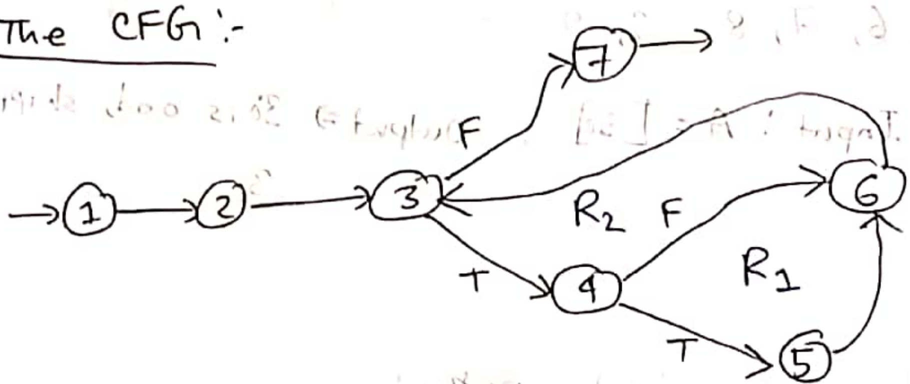



∴ Cyclomatic complexity = Number of Regions + 1 = 3 + 1
= 4

## Code 2 :-

```
int sum = 0;                                   1
          2          3              6
for ((int i=0); (i< A.length); (i++))
{
    if (A[i] % 2 == 0)                         4
    {
        sum += A[i]; }                         5
    }
}
System.out.println(sum);                       7
```

∴ The CFG :-




∴ The cyclomatic complexity = 2+1 = 3

So, the Code 2 has a better cyclomatic complexity than the Code 1.

For Code 1, independent paths :-

① 1, 2, 3, 9

Test :- Input : A = [ ] , Output = 0

② 1, 2, 3, 4, 5, 8, 3, 9

Test :- Input : A = [2'0] , Output = 2

③ 1, 2, 3, 4, 6, 8, 3, 9

Test :- Input :- A = [3'5] , Output = 0

④ 1, 2, 3, 4, 6, 7, 8, 3, 9

Test :- Input : A = [3'0] , Output ⇒ 3'0 is odd, skipping ...
3

For code 2, independent paths :-

(i) 1, 2, 3, 7

Test :- input :- A = [ ] , output = 0

(ii) 1, 2, 3, 4, 6, 3, 7

Test :- input :- A = [3'0] , output = 0

(iii) 1, 2, 3, 4, 5, 6, 3, 7

Test :- Input :- A = [2'0] , output = 2

—o—×—o—

## Ans to the ques no:- 9

**Inheritence Tree :-**

Level 0 ⟶ Animal          FlyingThing
fly, dance

Level 1 ⟶ Bird
~~fly~~ Fly
dance
sound, hunt

For class Bird,

$$SIX = \frac{NMO \times DIT}{NMO + NMI + NMA} \times 100\%$$

$$= \frac{1 + 1}{1 + 1 + 2} \times 100\%$$

$$= \frac{2}{4} \times 100\%$$

$$= 50\%$$

(Ans)

— o — x — o —