Q1: Test Cases for Previous Date Program

1. Equivalence Partitioning Test Cases

Valid Classes:

- Valid dates within the range:
    - (1, 1, 1900) → Expected Output: (31, 12, 1899)
    - (31, 12, 2015) → Expected Output: (30, 12, 2015)
    - (29, 2, 2012) → Expected Output: (28, 2, 2012)
-

Invalid Classes:

- Invalid month:
    - (0, 1, 2000) → Expected Output: An Error message
    - (13, 1, 2000) → Expected Output: An Error message
- Invalid day:
    - (32, 1, 2000) → Expected Output: An Error message
- Invalid year:
    - (15, 3, 1899) → Expected Output: An Error message
    - (15, 3, 2016) → Expected Output: An Error message

Equivalence Partitioning Test Cases Summary:

Tester Action and Input Data	**Expected Outcome**
(1, 1, 1900)	(31, 12, 1899)
(31, 12, 2015)	(30, 12, 2015)
(29, 2, 2012)	(28, 2, 2012)

(0, 1, 2000)	An Error message
(32, 1, 2000)	An Error message
(13, 1, 2000)	An Error message
(15, 3, 1899)	An Error message
(15, 3, 2016)	An Error message

Boundary Value Analysis Test Cases

Boundary Conditions:

- Valid edge cases:
  - (1,1,1900) → Expected Output: (31,12,1899)
  - (1,1,2015) → Expected Output: (31,12,2014)
  - (1,2,1900) → Expected Output: (31,1,1900)
  - (29 ,2 ,2015) → Expected Output: An Error message
  - (29 ,2 ,2016) → Expected Output: (28 ,2 ,2016)
  - (31 ,12 ,1900) → Expected Output: (30 ,12 ,1900)
  - (31 ,12 ,2015) → Expected Output: (30 ,12 ,2015)

Boundary Value Analysis Test Cases Summary:

Tester Action and Input Data	**Expected Outcome**
(1 ,1 ,1900)	(31 ,12 ,1899)

(1 ,1 ,2015)	(31 ,12 ,2014)
(1 ,2 ,1900)	(31 ,1 ,1900)
(29 ,2 ,2015)	An Error message
(29 ,2 ,2016)	(28 ,2 ,2016)
(31 ,12 ,1900)	(30 ,12 ,1900)
(31 ,12 ,2015)	(30 ,12 ,2015)

Q2: Programs

P1. Linear Search

The `linearSearch` function searches for a value v in an array$a$. If found it returns the index; otherwise returns -1.

```
int linearSearch(int v, int a[]) {
    int i = 0;
    while(i < a.length){
        if(a[i] == v)
            return i;
        i++;
    }
    return -1;
```

```
}
```

Test Cases for Linear Search

Tester Action and Input Data	Expected Outcome
(10,``)	0
(10,``)	-1
(10,[20,10])	1
(10,[20,-10])	-1

P2. Count Item

The `countItem` function counts occurrences of value v.

```
int countItem(int v,int a[]) {
    int count = 0;
    for(int i = 0; i < a.length; i++) {
        if(a[i] == v)
            count++;
    }
    return count;
}
```

Test Cases for Count Item

Tester Action and Input Data	Expected Outcome

`(10,``)`	`1`
`(10,``)`	`0`
`(10,[10,-10,-10])`	`1`

P3. Binary Search

The `binarySearch` function searches for v.

```
int binarySearch(int v,int a[]) {
    int lo = 0;
    int hi = a.length-1;
    while(lo <= hi){
        int mid = lo + hi /2;
        if(v == a[mid])
            return mid;
        else if(v < a[mid])
            hi = mid-1;
        else
            lo = mid+1;
    }
    return -1;
}
```

Test Cases for Binary Search

Tester Action and Input Data	Expected Outcome

(10,``)	0
(10,``)	-1
(10,[20,-10])	-1

P4. Triangle Classification

The `triangle` function classifies triangle types.

```
int triangle(int a,int b,int c){
    if(a >= b+c || b >= a+c || c >= a+b)
        return INVALID;
    if(a == b && b == c)
        return EQUILATERAL;
    if(a == b || a == c || b == c)
        return ISOSCELES;
    return SCALENE;
}
```

Test Cases for Triangle Classification

Tester Action and Input Data	Expected Outcome
(3,3,3)	EQUILATERAL
(3,3,4)	ISOSCELES

`(3,4,5)`	`SCALENE`

P6. Additional Questions on Triangle Classification Program

a. Identify Equivalence Classes

- Equilateral: All sides equal.
- Isosceles: Two sides equal.
- Scalene: All sides different.
- Invalid: Sides do not form a triangle.

b. Identify Test Cases Covering Equivalence Classes

Test Cases Summary

- `(3 ,3 ,3)` → EQUILATERAL
- `(3 ,3 ,4)` → ISOSCELES
- `(3 ,4 ,5)` → SCALENE
- `(1 ,2 ,3)` → INVALID

c. Boundary Condition A + B > C Case

Test case to verify boundary:

- `(3 +4 >7)` should yield INVALID.

d. Boundary Condition A = C Case

Test case to verify boundary:

- `(3 = C)` should yield ISOSCELES.

e. Boundary Condition A = B = C Case

Test case to verify boundary:

- `(4 =4=4)` should yield EQUILATERAL.

f. Boundary Condition $A^2 + B^2 = C^2$ Case

Test case to verify boundary:

- `(3² +4²=5²)` should yield SCALENE.

g. Non-Triangle Case Identification

Test case to explore the boundary:

- `(10 +2 <3)` should yield INVALID.

h. Non-positive Input Points

Test cases for non-positive input:

- `(-3 ,-4 ,-5)` should yield INVALID.