**Task1:**

**A)**

The error in i < arr.length-1

Solution:i<-arr.length-1;

B) no case will excecute fault without resulting to an error as in excecute the last index will not be accessed

-----------------------------------

**C)** last index is 0 so there is no failure

arr = [4,4,4,0]

expected: 12/4 =3

actual: 12/4 =3

**D)** if the last index is any number except 0 will be failure

Arr=[4,4,4,4]

Excepted:16/4=4

Actual:12/4=3

**E)** last value will not be added to the sum because I in for loop will not refer to the last value

Input:[90.5,-65.0,72.25]

Expected:32.58

Actual:8.5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Methods | Parameter | Return | Values | Exceptions | Char ID | characteristics | Covered by |
| adjacentHash() | String Hash, Direction dir | string | Adjacent hash | java.lang.  IllegalArgumentException  ,java.lang.NullPointerException | C1  C2 | Hash  direction |  |
| adjacentHash() | String Hash,direction dir,  int steps | string | hash at position in direction a number of hashes away | java.lang.  IllegalArgumentException,  java.lang.StringIndexOutOfBoundsException | C3 | Steps | C1,C2 |
| Right() | String hash | string | hash on right of given hash | java.lang.  IllegalArgumentException, java.lang.StringIndexOutOfBoundsException |  |  | C1 |
| left() | String Hash | string | hash on right of given hash | java.lang.  IllegalArgumentException, java.lang.StringIndexOutOfBoundsException |  |  | C1 |
| top() | String Hash | string | hash on right of given hash | java.lang.  IllegalArgumentException, java.lang.StringIndexOutOfBoundsException |  |  | C1 |
| Bottom() | String Hash | string | hash on right of given hash | java.lang.  IllegalArgumentException, java.lang.StringIndexOutOfBoundsException |  |  | C1 |
| Neighbours() | String Hash | List<String> | A list neighbouring hashes | java.lang.  IllegalArgumentException, java.lang.StringIndexOutOfBoundsException |  |  | C1 |
| coverBoundingBox() | Double topLeftLat, Double topLeftLon, Double bottomRightLat, Double bottomRightLon | coverage | Returns the result of cover Bounding Box Max Hashes with a max Hashes value of the default value(12) | java.lang.  IllegalArgumentException | C4  C5  C6  C7 | topLeftLat  TopLeftLon  bottomRightLat  bottomRightLon |  |
| coverBoundingBox() | Double topLeftLat, Double topLeftLon, Double bottomRightLat, Double bottomRightLon, Int length | coverage | Returns the hashes of given length that are required to cover the given bounding box | java.lang.  IllegalArgumentException | C8 | Length | C4,C5,C6,C7 |
| coverBounding  Boxmaxhashes() | Double topLeftLat, Double topLeftLon,  Double bottomRightLat,Double bottomRightLon,  Int maxhashes | coverage | Hashes required to cover given boundary box,null | java.lang.  IllegalArgumentException | C9 | MaxHashes length | C4,C5,C6,C7 |
| Decode Hash() | String geohash | LatLong | Latlong points | java.lang.NullPointerException |  |  | C1 |
| Encode hash() | Double Latitude, Double longitude | string | Hash at given point of default length |  | C10  C11 | Latitude length  Longitude length |  |
| Encode hash() | Double Latitude, Double longitude  ,int length | string | Hash at given point of default length | java.lang.  IllegalArgumentException |  |  | C8,C10,C11 |
| Encode hash() | LatLong P | string | Geohash of length for given point |  |  |  | C10 |
| Encode hash() | LatLong P, int length | string | Geohash of given length at given point |  |  |  | C8,C10 |
| GridAsString() | String Hash, int fromRight, int Frombottom, int Toright,int Tobottom | string | String of lines of hashes | java.lang.NullPointerException | C12  C13  C14  C15 | fromRight  fromBottom  ToRight  ToBottom | C1 |
| GridAsString() | String Hash,int fromRight, Int frombottom, int toright, int tobottom  Set<string> highlightThese | string | String representation  Of grid | java.lang.NullPointerException | C16 | HighlightThese | C1,C12,C13,C14,C15 |
| GridAsString() | String Hash, int Size, Set<string>  highlightThese | string | String representation  Of grid | java.lang.NullPointerException | C17 | Size | C1,C16 |
| Hashcontains() | String Hash ,Double lat, Double Lon | Boolean | True, false |  |  |  | C1,C10,C11 |
| hashLengthToCover  BoundingBox() | Double topLeftLat , Double topLeftLon, Double bottomRightLat, Double bottomRightLon | int | Length of the hash |  |  |  | C4,C5,C6,C7 |
| heightDegree() | Int n | double | Returns height in degrees | java.lang.ArrayIndexOutOfBoundsException | C18 | N |  |
| Widthdegree() | Int n | double | Returns width in degrees | java.lang.ArrayIndexOutOfBoundsException |  |  | C18 |

**Identify Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Characteristics | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 | C16 | C17 | C18 |
| Methods |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| adjacentHash() | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| AdjacentHash() | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Right() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| left() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| top() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| bottom() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Neighbours() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CoverBoundingBox() |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |
| CoverBoundingBox() |  |  |  | X | X | X | X | X |  |  |  |  |  |  |  |  |  |  |
| CoverBoundingBoxMaxHashes() |  |  |  | X | X | X | X |  | X |  |  |  |  |  |  |  |  |  |
| decoderHash() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| encoderHash() |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |
| encoderHash() |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |  |  |
| encoderHash() |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| encoderHash() |  |  |  |  |  |  |  | X |  | X |  |  |  |  |  |  |  |  |
| GridAsString() | X |  |  |  |  |  |  |  |  |  |  | X | X | X | X |  |  |  |
| GridAsString() | X |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X |  |  |
| GridAsString() | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |
| HashContains() | X |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |
| HashLengthToCoverBoundingBox() |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |
| heightDegree() |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |
| WidthDegree() |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |

**Characteristics Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Characteristics | B1 | B2 | B3 |
| C1 | Hash = null | Hash = Not null |  |
| C2 | Direction = null | Direction = Not null |  |
| C3 | Steps > 0 | Steps < 0 |  |
| C4 | TopLeftLat >= bottomRightLat | TopLeftLat <= bottomRightLat |  |
| C5 | TopLeftLon > 0 | TopLeftLon < 0 |  |
| C6 | TopLeftLat <= bottomRightLat | TopLeftLat >= bottomRightLat |  |
| C7 | bottomRightLon > 0 | bottomRightLon < 0 |  |
| C8 | 0 <= length <= 12 | Length < 0 | Length > 12 |
| C9 | MaxHash > 0 | MaxHash < 0 |  |
| C10 | -90 <= latitude <= 90 | Latitude > 90 | Latitude > -90 |
| C11 | -180<= longitude <= 180 | Longitude > 180 | Longitude > -180 |
| C12 | FromRight > 0 | FromRight < 0 |  |
| C13 | FromBottom > 0 | FromBottom < 0 |  |
| C14 | toRight > 0 | ToRight < 0 |  |
| C15 | toBottom > 0 | ToBottom < 0 |  |
| C16 | HighlightThese = null | HighlightThese = not null |  |
| C17 | Size > 0 | Size < 0 |  |
| C18 | N > 0 | N < 0 |  |

**Partitions Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | Characteristics | Test Requirements | Infeasible | # of requirements |
| adjacentHash() | C1,C2 | {  [C1.b2,C2.b2]  [C1.b1,C2.b2]  [C1.b2,C2.b1]  } | {  [C1.b1,C2.b2]  [C1.b2,C2.b1]  }  Hash and direction cannot be null | 1 |
| adjacentHash() | C1,C2,C3 | {  [C1.b2,C2.b2,C3.b1]  [C1.b1,C2.b2,C3.b1]  [C1.b2,C2.b1,C3.b1]  [C1.b2,C2.b2,C3.b2]  } | {  [C1.b1,C2.b2,C3.b1]  [C1.b2,C2.b1,C3.b1]  }  Hash and direction cannot be null | 1 |
| Right() | C1 | {  [C1.b1]  [C2.b2]  } | {  [C2.b1]  }  Hash cannot be null | 1 |
| left() | C1 | {  [C1.b1]  [C2.b2]  } | {  [C2.b1]  }  Hash cannot be null | 1 |
| top() | C1 | {  [C1.b1]  [C2.b2]  } | {  [C2.b1]  }  Hash cannot be null | 1 |
| Bottom() | C1 | {  [C1.b1]  [C2.b2]  } | {  [C2.b1]  }  Hash cannot be null | 1 |
| Neighbours() | C1 | {  [C1.b1]  [C2.b2]  } | {  [C2.b1]  }  Hash cannot be null | 1 |
| coverBoundingBox() | C4,C5,C6,C7 | {  [C4.b1,C5.b1,C6.b2,C7.b1]  [C4.b2,C5.b1,C6.b2,C7.b1]  [C4.b1,C5.b2,C6.b2,C7.b1]  [C4.b1,C5.b1,C6.b1,C7.b1]  [C4.b1,C5.b1,C6.b2,C7.b2]  } | {  [C4.b2,C5.b1,C6.b2,C7.b1]  [C4.b1,C5.b1,C6.b1,C7.b1]  }  topLeftLat must be >= BottomLeftLat | 3 |
| coverBoundingBox() | C4,C5,C6,C7,C8 | {  [C4.b1,C5.b1,C6.b2,C7.b1,C8.b1]  [C4.b2,C5.b1,C6.b2,C7.b1,C8.b1]  [C4.b1,C5.b2,C6.b2,C7.b1,C8.b1]  [C4.b1,C5.b1,C6.b1,C7.b1,C8.b1]  [C4.b1,C5.b1,C6.b2,C7.b2,C8.b1]  [C4.b1,C5.b1,C6.b2,C7.b1,C8.b2]  [C4.b1,C5.b1,C6.b2,C7.b1,C8.b2]  } | {  [C4.b2,C5.b1,C6.b2,C7.b1,C8.b1]  [C4.b1,C5.b1,C6.b1,C7.b1,C8.b1]  [C4.b1,C5.b1,C6.b2,C7.b1,C8.b2]  [C4.b1,C5.b1,C6.b2,C7.b1,C8.b2]  }  topLeftLat cannot be >= bottomRightLat and length cannot be negative and exceed 12 | 3 |
| coverBoundingBoxmaxhashes() | C4,C5,C6,C7,C9 | {  [C4.b1,C5.b1,C6.b2,C7.b1,C9.b1]  [C4.b2,C5.b1,C6.b2,C7.b1,C9.b1]  [C4.b1,C5.b2,C6.b2,C7.b1,C9.b1]  [C4.b1,C5.b1,C6.b1,C7.b1,C9.b1]  [C4.b1,C5.b1,C6.b2,C7.b2,C9.b1]  [C4.b1,C5.b1,C6.b2,C7.b1,C9.b2]  } | {  [C4.b2,C5.b1,C6.b2,C7.b1,C9.b1]  [C4.b1,C5.b1,C6.b1,C7.b1,C9.b1]  [C4.b1,C5.b1,C6.b2,C7.b1,C9.b2]  }  toLeftLat cannot be >= bottomRightLat and maxHashes cannot be negative | 3 |
| Decode Hash() | C1 | {  [C1.b1]  [C1.b2]  } | {  [C1.b1]  }  Hash cannot be null | 1 |
| Encode hash() | C10,C11 | {  [c10.b1,C11.b1]  [C10.b2,C11.b1]  [C10.b3,C11.b1]  [C10.b1,C11.b2]  [C10.b1,C11.b3]  } | {  [C10.b2,C11.b1]  [C10.b3,C11.b1]  }  Latitude cannot exceed 90 and -90 | 3 |
| Encode hash() | C10,C11,C12 | {  [C10.b1,C11.b1,C12.b1]  [C10.b2,C11.b1,C12.b1]  [C10.b3,C11.b1,C12.b1]  [C10.b1,C11.b2,C12.b1]  [C10.b1,C11.b3,C12.b1]  [C10.b1,C11.b1,C12.b2]  } | {  [C10.b2,C11.b1,C12.b1]  [C10.b3,C11.b1,C12.b1]  }  Latitude cannot exceed 90 and -90 | 4 |
| Encode hash() | C10 | {  [C10.b1]  [C10.b2]  [C10.b3]  } | {  [C10.b2]  [C10.b3]  }  Latitude cannot exceed 90 and -90 | 1 |
| Encode hash() | C8,C10 | {  [C8.b1,C10.b1]  [C8.b2,C10.b1]  [C8.b3,C10.b1]  [C8.b1,C10.b2]  [C8.b1,.=C10.b3]  } | {  [C8.b2,C10.b1]  [C8.b3,C10.b1]  [C8.b1,C10.b2]  [C8.b1,C10.b3]  }  Length cannot exceed size 12 and latitude cannot exceed 90 and -90 | 1 |
| GridAsString() | C1,C12,C13,C14,C15 | {  [C1.b2,C12.b1,C13.b1,C14.b1,C15.b1]  [C1.b1,C12.b1,C13.b1,C14.b1,C15.b1]  [C1.b2,C12.b2,C13.b1,C14.b1,C15.b1]  [C1.b2,C12.b1,C13.b2,C14.b1,C15.b1]  [C1.b2,C12.b1,C13.b1,C14.b2,C15.b1]  [C1.b2,C12.b1,C13.b1,C14.b1,C15.b2]  } | {  [C1.b1,C12.b1,C13.b1,C14.b1,C15.b1]  }  Hash cannot be null | 5 |
| GridAsString() | C1,C12,C13,C14,C15,C16 | {  [C1.b2,C12.b1,C13.b1,C14.b1,C15.b1,C16.b2]  [C1.b1,C12.b1,C13.b1,C14.b1,C15.b1,C16.b2]  [C1.b2,C12.b2,C13.b1,C14.b1,C15.b1,C16.b2]  [C1.b2,C12.b1,C13.b2,C14.b1,C15.b1,C16.b2]  [C1.b2,C12.b1,C13.b1,C14.b2,C15.b1,C16.b2]  [C1.b2,C12.b1,C13.b1,C14.b1,C15.b2,C16.b2]  [C1.b2,C12.b1,C13.b1,C14.b1,C15.b1,C16.b1]  } | {  [C1.b1,C12.b1,C13.b1,C14.b1,C15.b1,C16.b2]  [C1.b2,C12.b1,C13.b1,C14.b1,C15.b1,C16.b1]  }  Hash cannot be null and highlightThese cannot be null | 5 |
| GridAsString() | C1,C16,C17 | {  [C1.b2,C16.b2,C17.b1]  [C1.b1,C16.b2,C17.b1]  [C1.b2,C16.b1,C17.b1]  [C1.b2,C16.b2,C17.b2]  } | {  [C1.b1,C16.b2,C17.b1]  [C1.b2,C16.b1,C17.b1]  [C1.b2,C16.b2,C17.b2]  }  Hash cannot be null and highlightThese cannot be null and size shouldn’t be -ve | 1 |
| Hashcontains() | C1,C10,C11 | {  [C1.b2,C10.b1,C11.b1]  [C1.b1,C10.b1,C11.b1]  [C1.b2,C10.b2,C11.b1]  [C1.b2,C10.b3,C11.b1]  [C1.b2,C10.b1,C11.b2]  [C1.b2,C10.b1,C11.b3]  } | {  [C1.b1,C10.b1,C11.b1]  [C1.b2,C10.b2,C11.b1]  [C1.b2,C10.b3,C11.b1]  }  Hash cannot be null and latitude cannot exceed 90 and -90 | 3 |
| heightDegree() | C18 | {  [C18.b1]  [C18.b2]  } | {  [C18.b2]  }  N cannot be negative value | 1 |
| Widthdegree() | C18 | {  [C18.b1]  [C18.b2]  } | {  [C18.b2]  }  N cannot be negative value | 1 |

**Base Coverage test Requirements**