Yang Xi Gao

a.

```
     for(int i = 0; i < given.length; i++){  //n

            for(int j = i+1; j < given.length; j++){ //(n(n+1)/2)

             if (given[i] + given[j] == givenSum){ //(n(n+1)/2)
                 return given[i] + " + " + given[j] + " = " +
givenSum + ", " + "true"; //1
           }
                 }


        }
     return "false";
```

Total number of operations: n^2+3n+2
Time complexity: O(n^2)

```
b. for(int i = 0; i < given.length; i++){ //n


         for(int j = i+1; j < given.length; j++){ //n(n+1)/2


           for(int k = j+1; k < given.length; k++){ //n(n+2)/2

               if (given[i] + given[j] + given[k] == key){ //n(n+2)/2

                return given[i] + "+" + given[j] + "+" + given[k] +
"=" + key + ", " + "true"; //1
               }
           }
         }
     }
     return "false";
```

Total Number of operations: $2n^2$+5n+2
Time complexity: O(n^3)

```
C. for (int i = 0; i < n; i++) { //n
           for (int j = 0; j < n; j++) { //n^2
                double sum = 0; //n^2
           for (int k = 0; k < n; k++) { //n^3
                sum += a[i][k] * b[k][j]; //n^3
           }
           c[i][j] = sum; //n^2
     }
}
```

Total Number of Operations: $2n^3+2n^2+n$

Time complexity: $O(n^3)$

Space complexity: $O(n^2)$