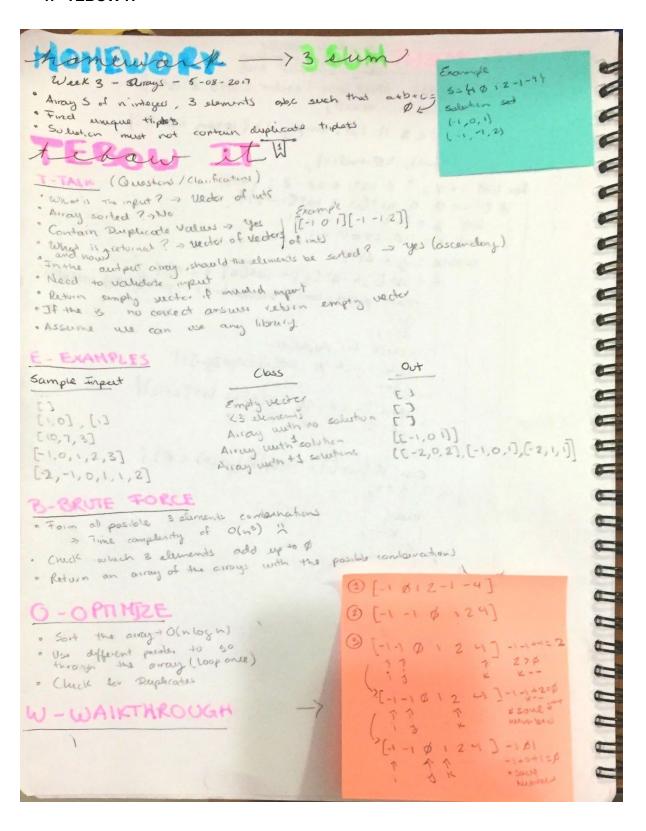
1. TEBOW IT



```
J-IMPLEMENT # moducle vector

# moducle vector

way names pour std.
vector & vector Lint>> three sum (wester Zints &list) (so
                          result;
    veder (veder cint >>
    if (list. size () 2 3 11 list. empty ()) { letvin result :}
      soit (list begin !), 1,57. end (1);
      for (int i = d; i < list. size -2; j++){
        if ( == 0 11 hist(i) > hist(i-1)){
            it i = f tri
           int k = list. size(1-1;
            while ( j < x) {
                 if ( list [1] + list [8] + list[x] == 0) {
                     result. push - back (vector cont > ( 1 10+0), 115+6), 115+64))
                     jet,
                     11 (heck for deplicate)
                     while (jex 22 listEs] == list Ey-1) }
                     while ( j cx 22 list[x] == list[x+1])}
                     } K--;
                    else it ( 10+ E) + 10+ E) + 10+ Ex) < 0) {
                    else
                 resent;
          cetur
```

2. CODE IMPLEMENTATION

#include<algorithm>
#include<vector>
using namespace std;

```
class Solution {
public:
   vector<vector<int> > threeSum(vector<int> &list) {
      vector<vector<int> > result;
      if(list.size()< 3) return result;</pre>
                 sort(list.begin(), list.end());
                 for (int i = 0; i < list.size() - 2; i++) {
         if(i == 0 || list[i] > list[i-1]) {
            int j = i + 1;
            int k = list.size() - 1;
            while (j < k) {
              if (list[i] + list[j] + list[k] == 0){
                  result.push_back(vector<int> ({ list[i], list[j], list[k] }));
                 j++;
                  k--;
                  //handle duplicate here
                  while(j < k \&\& list[j] == list[j-1])
                     j++;
                  while(j < k \&\& list[k] == list[k+1])
                     k--;
              }
              else if(list[i]+list[j]+list[k] < 0){
                 j++;
              }
               else{
                  k--;
              }
           }
        }
                 }
                 return result;
        }
};
```