Greedy Assignment1

Q1. Find the minimum number of fibonacci numbers whose sum is K [Leetcode 1414];

Code;-
```
class Solution {

public:
    int findMinFibonacciNumbers(int k) {
        int c=1,c1=1;
        vector<int>v;
        while(c<=k){
           v.push_back(c);
            int t=c+c1;
            c1=c;
            c=t;
        }
        int j=v.size()-1;
        int count=0;
        while(k){
            if(v[j]>k)j--;
            else{
                k-=v[j];
                count++;
            }
        }
        return count;
    }
};
```

Q2. K items with maximum sum [Leetcode 2600];

Code:-
```
class Solution {

public:
    int kItemsWithMaximumSum(int numOnes, int numZeros, int numNegOnes, int k)
{
        int sum=0;
        int t=min(k,numOnes);
        k-=t;
        sum+=t;
        t=min(k,numZeros);
        k-=t;
        t=min(k,numNegOnes);
       // k-=t;
        sum-=t;
        return sum;
    }
};
```

Q3. Gas station [Leetcode 134];

Code;-
```
class Solution {

public:
    int canCompleteCircuit(vector<int>& gas, vector<int>& cost) {
        int c=0,g=0;
        int n=gas.size();
        int cur=0;
        int star=0;
        for(int i=0;i<gas.size();i++)g+=gas[i];
        for(int i=0;i<cost.size();i++)c+=cost[i];
        if(c>g)return -1;
        else{
            for(int i=0;i<n;i++){
                if(cur<0){
                    cur=0;
                    star=i;

                }
                cur+=gas[i]-cost[i];
            }
        }
        return star;
    }
};
```

Q4. Make array zero by subtracting equal amounts [Leetcode 2357];

Code:-
```
class Solution {

public:
    int minimumOperations(vector<int>& nums) {
       set<int>s;
        for(auto x:nums){
            if(x>0)s.insert(x);
        }
        return (int)s.size();
    }
};
```