int cmp(const void \*a,const void \*b){

    return \*(int\*)a -\*(int\*)b;

}

/\*\*

 \* Return an array of arrays of size \*returnSize.

 \* The sizes of the arrays are returned as \*returnColumnSizes array.

 \* Note: Both returned array and \*columnSizes array must be malloced, assume caller calls free().

 \*/

int\*\* threeSum(int\* nums, int numsSize, int\* returnSize, int\*\* returnColumnSizes){

    int\*\* ans=malloc(sizeof(int\*)\*1);

    int c=0;

    \* returnColumnSizes=NULL;

    \* returnColumnSizes=realloc(\*returnColumnSizes,sizeof(int)\*(c+1));

    \* returnSize=c;

    int i=0;

    int k=i+1;

    int j=numsSize-1;

    //int check=0;

    //int mid=k;

    //printf("%d \n",numsSize);

    qsort(nums,numsSize,sizeof(int),cmp);

    //for(int m = 0 ; m < numsSize ; m++){

    //    printf("%d ",nums[m]);

    //}

    //printf("\n");

    while(1){

        int sum=nums[i]+nums[k]+nums[j];

        //printf("%d, nums[i=%d]=%d + nums[j=%d]=%d + nums[k=%d]=%d\n",sum,i,nums[i],j,nums[j],k,nums[k]);

        //printf("check2=%d\n",check);

        if( sum == 0){//確認是否sum==0;

        //printf("check1=%d\n",check);

            if(c>=1){

                int\*\* ans2=realloc(ans,sizeof(int\*)\*(c+1));

                ans=ans2;

            }

            \* returnColumnSizes=realloc(\*returnColumnSizes,sizeof(int)\*(c+1));

            \* returnSize=c+1;

            ans[c]=malloc(sizeof(int)\*3);

            ans[c][0]=nums[i];

            ans[c][1]=nums[k];

            ans[c][2]=nums[j];

            (\* returnColumnSizes)[c]=3;

            c++;

            do{

            j--;

            k++;

            //printf("check3=%d j=%d k=%d\n",check,j,k);

            }while(  j > 1 && k < numsSize-2 && nums[j]==nums[j+1] && nums[k]==nums[k-1] );

        }else if( sum > 0){

                j--;

        }else if( sum < 0){

                k++;

        }

        if( j <= k){

           //printf("check4=%d i=%d j=%d k=%d\n",check,i,j,k);

            do{

                i++;

                k=i+1;

                j=numsSize-1;

            //printf("check5=%d i=%d j=%d k=%d\n",check,i,j,k);

            }while(  i < numsSize-2 && nums[i]==nums[i-1] );

        }

        if( k==j ){break;}

    }

    return ans;

}