1. I firstly set an empty list ’org\_list’, and in order to be more convenient to do the further coding, i introduced a which is exactly same as input\_str, then i utilized the ‘while’ circuit : whenever the length of a is not zero, a continually equals to the first (len(a)-1) string, meanwhile i add the new a in every tries of the circuit to the org\_list. Finally return the org\_list. print(input\_nput\_put\_ut\_t('hi! Mr.Adam, this is my homework’)) is just for checking the code.
2. In order to add two vectors in two lists, i define the function, then set an org\_list and use a ‘for’ circuit to try every amount of a in v1, so that sum1 is equal to the number a in v1 and number a in v2. thus i got the different values of sum of the elements in two lists in the same position. Finally, return the original list.

3. In order to find the slices in two different list with the same sum, i define the function: firstly, it i use all together four ‘for’ circuit, to construct the beginofa, endofa, beginofb, endofb. Then l1 is the every slices of list1 and l2 is defined the every slices we can find in list2. Finally, i use the while circuit to define the final restriction which is when l1 and l2 is not empty, the chosen slices are

when sum of l1 is equal to sum of l2. Finally return the four integers.( for the record, endofa - 1 is because in python, l1[3,9] means the number 3 elements in the list to number 8 in the list instead of number 9, so i need to subtract 1). print(find\_equal\_sum\_slice([7,8,90],[6,9,91])) is for checking my code.

4. in order to sort a sequence of numbers, i set an org\_list which is empty and make it more convenient, a for v1. then i utilized the ‘for’ circuit to try every number in the list. i add the minimum element quantitively in a to the original empty list, then delete that element. By recycling doing that, is successfully get the new org\_list. print(basic\_sort([3,6,2,4,5,12,9,7,1,8])) is for checking my code.