Made by: Ebrahim Abdelghfar

Problem 1

```
#made by Ebrahim Abdelghfar
#made by python 3.9.7
#equiphontaic program
def equiphontaic(str1,str2,length):
  condition=False
  counter=0
  index1=0
  for i  in range(length):
    for j in range(length):
      if(str1[i]==str2[j]):
        counter+=1
        break
  if(counter==length):
    condition=True
    print("equaphontics("+str(condition)+")")
  else:
    condition=False
    print("equaphontics("+str(condition)+")")

while(True):
    word1=str.lower(input("please enter the first word :"))
    word2=str.lower(input("please enter the seconed word :"))
    length_word_1=len(word1)
    length_word_2=len(word2)
    if (length_word_1==length_word_2):
        equiphontaic(word1,word2,length_word_1)

    else:
        print("not equiphontaic please enter another word")
    choice=str.capitalize(input("enter ""Y"" if you wish to continue or ""N"" to
exit :"))
    if(choice=="N"):
      break
    else:
       continue
```

Problem 2:

```
#made by Ebrahim Abdelghfar
#made by python 3.9.7
# modified bubble sorting
def cmp(a, b):
    return (a >= b) - (a <= b)

def modified_sorting(list,length_of_list): #comprison and swap function of the
array
    Counter_1 =0
    for i in range(length_of_list):
        temp=list
        for j in range (length_of_list-2):
            if(list[j]>=list[j+2]):
               list[j],list[j+2]=list[j+2],list[j]
            elif(j==(length_of_list-2)):
                if(list[j]>=list[j+1]):
                    list[j],list[j+2]=list[j+2],list[j]
                    break
            else:
                continue
        Counter_1+=1
        print(Counter_1)
        if(cmp(list,temp)==1):  #this condition is to decrease complexity of
algorithms by break the un necessary loop
            break
        else:
            continue #continue the loop when the next loop is nessesary


while(True):
    array=[]#defining array
    list_length=int(input("please enter the number of items you will add :"))
    for i in range (list_length):
        array.append(float(input("the item with index ("+str(i)+") =")))
    print("the list you enter ="+str(array))
    modified_sorting(array,list_length)
    print("the list after sorting="+str(array))
    choice=str.capitalize(input("enter ""Y"" if you wish to continue or ""N"" to
exit :"))
    if(choice=="N"):
      break
    else:
       continue
```

Self-driving Car module:

ROS:

ROS is acronym for "robot operating system" that used in many robotics nowadays in addition containing.

A lot of software libraries and tool that facilitate programming a complex robot in addition to the contributor a developer that develop and create packages that facilitate programming robotics software. And it only installed on Linux based system such as Ubuntu. The workspace that used contain 3 main directory: 1-build 2-devel 3-src. It can be programmed by Python or C++ .

AirSim:

Airsim is a simulator build by unreal engine used to simulate Cars and drones so that it mainly used in AI research center, it can be used in different operating system.

SLAM:

SLAM is acronym for “Simultaneous localization and mapping”, most used types: 1-Fastslam 2-occupancy grid SLAM 3-graph SLAM. SLAM used in making map and localizing robot in the same moment it with in the map it mainly depend on particle filter on estimation process

LIDAR:

Lidar is acronym for “light detection and ranging” it’s working mechanism as follow it have a rotating mirror with (600 to 4800 RPM) that reflect a shiny laser beam and it depend on measuring flight time of each beam , it has significant aerodynamics forces, it has longer range and higher accuracy. It can be used in distinguishing between different object and draw maps

Navigation control:

It’s used to in path planning to and determine the shortest distance between one point and it use A* algorithm in Path searching

Kalman filter:

It’s used in filter and noise remover but the sensor should follow a Gaussian distribution or it will not work, but EKF (extended kalman filter) can work with non-Gaussian sensors. They are used in localization and sensor fusion.

LaserNet:

The main sensor used is LIDAR, The LiDAR produces a cylindrical range image as it sweeps over the environment with a set of lasers. The horizontal resolution of the image is determined by the rotaotion speed and laser pulse rate, and the vertical resolution is determined by the number of lasers