```python
# 1.定义一个函数，该函数可以实现控制台输入，最终返回一个int类型的正整数
# 解析：如何将字符串转换为int类型的正整数
def back_r_int():
    while True:
        num = input('num: ')
        if num.isdigit():
            return int(num)

res = back_r_int()
print(res)
```

```python
# 2.定义一个函数，该函数可以实现用户录入，最终返回一个int类型的负整数
# 解析：1.只能有一个'-'且以它开头；2.按'-'拆分得到的一定是长度为2的列表，且后面的数是正整数
def back_l_int():
    while True:
        num = input('num: ')
        if num.startswith('-') and num.count('-') == 1:
            r_part = num.split('-')[1]
            if r_part.isdigit():
                return int(num)
res = back_l_int()
print(res)
```

```python
# 3.定义一个函数，实现传入一个数或是字符串，返回值是 是否 是可转换为整数类型的数据
# 解析：运用1，2的逻辑判断
def is_int(num):
    if isinstance(num, int):
        return True
    if isinstance(num, str):
        if num.isdigit():
            return True
        if num.startswith('-') and num.count('-') == 1:
            r_part = num.split('-')[1]
            if r_part.isdigit():
                return True
    return False

print(is_int('-10'))
```

```python
# 4.定义一个函数，实现传入一个整型数字，判断并直接打印该数字是否是奇数还是偶数
# 解析：解决奇数、偶数的概念即可
def is_single(num):
    if num % 2 == 1:
        print('奇数')
    else:
        print('偶数')

is_single(10)
```

```python
# 5.定义一个函数，实现判断传入数据的类型，并直接打印其类型
# 解析：如何判断数据的类型
type_map = {
    int: '整型',
    bool: '字符串',
    'float': '浮点型',
    'complex': '复数',
    'list': '列表',
    'dict': '字典',
    'set': '集合',
    'tuple': '元组',
}
def data_type(data):
    if isinstance(data, bool):
        print('type - bool')
        return
    if isinstance(data, int):
        print('type - int')
        return
    if isinstance(data, float):
        print('type - float')
        return
    if isinstance(data, complex):
        print('type - complex')
        return
    if isinstance(data, str):
        print('type - str')
        return
    if isinstance(data, list):
        print('type - list')
        return
    if isinstance(data, tuple):
        print('type - tuple')
        return
    if isinstance(data, dict):
        print('type - dict')
        return
    if isinstance(data, set):
```

```
        print('type - set')
        return

ls = [1, '1', 3.14, True, [], {}, {1,}, (1,), 4+5j]
for v in ls:  # 丢进去各种功能，检测我们的功能是否能检查这些类型
    data_type(v)
```

```
# 6.定义一个函数，实现可以重复录入键盘信息，当用户输入q或Q时退出，否则判断是否为可转换为整数类型的数据，可以的话输出该数是奇数还是偶数，否则直接输出该字符串
# 解析：要调用3，4题结果
def print_info():
    while True:
        num = input('num: ')
        if num == 'q' or num == 'Q':
            return
        res = is_int(num)  # 使用第3题函数判断是否是整数类型的数据
        if res:  # 是整数类型数字
            num = int(num)  # 转换为int类型
            is_single(num)  # 使用第4题函数打印奇数偶数
        else:
            print(num)
print_info()
```

```
# 7.定义一个函数，只要传入 "k1:v1,...,kn:vn" 格式的字符串，都可以将其转换为 {'k1':'v1',...,'kn':'vn'}
# 解析：字符串拆分与for循环迭代
def to_dic(string):
    dic = {}
    k_v_list = string.split(',')
    for k_v in k_v_list:
        k, v = k_v.split(':')
        dic[k] = v
    return dic

res = to_dic("name:owen,age:18,gender:male")
print(res)
```

```
# 8.定义一个函数，实现列表与元组类型的反转功能
# 解析：传入列表返回元组，传入元组返回列表
def toggle_list_tuple(target):
    if isinstance(target, tuple):
        return list(target)
    if isinstance(target, list):
        return tuple(target)
res = toggle_list_tuple([1, 2, 3])
print(res)
```

```python
# 9.定义一个函数，可以完成对list、tuple、dict、set四种类型数据的循环变量打印，不是这四种，则打印 "暂不支持该数据遍历"
# 解析：对数据类型做判断
def for_set(data):
    if isinstance(data, list) or isinstance(data, tuple) or isinstance(data, set):
        for v in data:
            print(v)
    if isinstance(data, dict):
        for k, v in data.items():
            print(k, v)
for_set([1, 2, 3])
```

```python
# 10.定义一个函数，实现对单列集合进行去重的功能
# 解析：单列集合有list、tuple、set，传入list、tuple、set，返回去重后的list、tuple、set，考虑可变与不可变类型的不同处理
def get_clear(data):
    if isinstance(data, set):
        return data
    temp_list = []
    for v in data:
        if v not in temp_list:
            temp_list.append(v)
    if isinstance(data, tuple):
        return tuple(temp_list)
    return temp_list

res = get_clear([3, 1, 2, 1])
print(res)
res = get_clear((3, 1, 2, 1))
print(res)
res = get_clear({3, 1, 2, 1})
print(res)
```

```python
# 11.定义一个函数，实现文件(不一定是文本文件)的跨文件夹的裁剪
# 解析：1.传入要读取的目标文件夹中的目标文件;2.在被告知的目标文件夹下复制成同名文件；3.调用os中删除文件的功能将原文件删除
import os
def move_file(file, folder):
    file_name = file.rsplit('\\', 1)[1]
    target_file = '%s\\%s' % (folder, file_name)
    with open(file, 'rb') as r, open(target_file, 'wb') as w:
        for line in r:
            w.write(line)
    os.remove(file)

move_file('D:\\temp.py', 'C:')
```

```python
# 拓展1：用函数实现判断一个字符串数据能否转换为正负小数
# 先考虑正小数，再在基础上考虑负小数，可以形成多个方法，形成函数的嵌套
# 正小数：只包含一个小数点，左右都是正整数
# 负小数：参考普通题的第2题结合正小数

def is_r_float(num_str):  # 正小数
    part_list = num_str.split('.')
    # 一个.，按.拆分的左右都是数字
    if len(part_list) == 2 and part_list[0].isdigit() and part_list[1].isdigit():
        return True
    return False

def is_l_float(num_str):  # 负小数
    # "-"开头，刨除"-"后是正整数
    if num_str.startswith('-') and is_r_float(num_str[1:]):
        return True
    return False
def is_float(num_str):
    # 正或负小数
    if is_l_float(num_str) or is_r_float(num_str):
        return True
    return False
```

```python
# 拓展2：实现汽车销售系统
'''
1）具有进货功能1，销售车辆功能2，展示所有库存功能3，展示销售总业绩功能4
2）用户输入0退出系统，输入提供的功能编号，完成对应的功能，否则重新输入，eg：2就进入销售车功能
3）车辆信息具有持久化(文件永久)存储功能，车辆有奔驰|宝马|奥迪三款
文件信息：
total.txt：就是记录了总销售额

car.txt：
宝马 120000 9
奔驰 150000 7
奥迪 100000 8

4）进货功能：选择进货的车型与数量，完成进货
5）售车功能：选择售出的车，有库存售出，更新销售总业绩，没有则进入进货功能
6）展示库存：显示所有车与库存两条信息即可
7）总业绩：显示总业绩金额即可

分析：要将total.txt与car.txt转换为合适的数据类型，操作完毕后同步到文件中即可
'''

def get_cars():
```

```
    cars_dic = {}
    with open('car.txt', 'r', encoding='utf-8') as f:
        data = f.read()
    if data:
        cars = data.split('\n')
        for car in cars:
            if car:
                name, price, count = car.split()
                cars_dic[name] = {'price': int(price), 'count': int(count)}
    return cars_dic

def update_cars(cars_dic):
    with open('car.txt', 'w', encoding='utf-8') as f:
        for car, info in cars_dic.items():
            line = car + ' '
            line += str(info['price']) + ' '
            line += str(info['count']) + '\n'
            f.write(line)


def get_grade():
    with open('total.txt', 'r', encoding='utf-8') as f:
        total = f.read()
    if total:
        return int(total)
    return 0

def update_grade(total):
    with open('total.txt', 'w', encoding='utf-8') as f:
        f.write(str(total))

def get_car():
    while True:
        car = input('宝马|奔驰|奥迪: ')
        if car in ['宝马', '奔驰', '奥迪']:
            return car

def get_count():
    while True:
        count = input('台数: ')
        if count.isdigit() and int(count) > 0:
            return int(count)


# 业绩
def grade(update=False, add=0):
    total = get_grade()
    if update:
        total += add
```

```
            update_grade(total)
        else:
            print('总业绩:', total)

# 库存
def hub():
    print('库存功能')
    cars_dic = get_cars()
    for car, info in cars_dic.items():
        print(car, info['count'])

# 售车
def sell():
    print('售车功能')
    cars_dic = get_cars()
    if not cars_dic:
        print('没有库存')
        stock()
    else:
        car = get_car()
        count = get_count()
        cars_dic = get_cars()
        if cars_dic[car]['count'] < count:
            print('库存不足')
            stock()
        else:
            cars_dic[car]['count'] -= count
            add = cars_dic[car]['price'] * count
            update_cars(cars_dic)
            grade(True, add)
            print('售车成功')
# 进货
def stock():
    print('进货功能')
    car = get_car()
    count = get_count()
    cars_dic = get_cars()
    cars_dic[car]['count'] += count
    update_cars(cars_dic)
    print('进货成功')

method_map = {
    '1': stock,
    '2': sell,
    '3': hub,
    '4': grade
}
def start():
    while True:
```

```
        cmd = input('''请输入:
1.进货
2.售车
3.库存
4.业绩
0.退出
>>>: ''')
        if cmd in method_map:
            method_map[cmd]()
        else:
            print('指令有误，重来')
start()
```