Task 1: Alternate Merge

```
def alternate_merge(head1, head2):
  dummy = Node(0)
  tail = dummy
  while head1 or head2:
    if head1:
      tail.next = head1
      tail = head1
      head1 = head1.next
    if head2:
      tail.next = head2
      tail = head2
      head2 = head2.next
  return dummy.next
```

Task 2: Word Decoder

```
def word_Decoder(head):
  length = 0
  temp = head
  while temp:
    length += 1
    temp = temp.next
  step = 13% length
  decoded_head = None
  temp =head
  position = 0
  while temp:
    if position % step == 0:
      new_node = Node(temp.elem)
      new_node.next = decoded_head
      decoded_head = new_node
    position += 1
    temp = temp.next
  dummy_head = Node(None)
  dummy_head.next = decoded_head

  return dummy_head
```

Task 3: ID Generator

```
def reverseList(head):
  prev = None
  curr = head
  while curr is not None:
    next_n = curr.next
    curr.next = prev
    prev = curr
    curr = next_n
```

```
  return prev

def idGenerator(head1, head2, head3):

    head1 = reverseList(head1)
    dummy = Node(0)
    tail = dummy

    curr2,curr3 = head2,head3
    while curr2 and curr3:
        sum = curr2.elem + curr3.elem
        tail.next = Node(sum % 10)
        tail = tail.next
        curr2 = curr2.next
        curr3 = curr3.next


    tail.next = None
    new_head = head1
    while new_head and new_head.next:
      new_head = new_head.next
    new_head.next = dummy.next

    return head1
```

Doubly Linked List

```
class Patient:
  def __init__(self, id, name, age, bloodgroup, next, prev):
    self.id = id
    self.name = name
    self.age = age
    self.bloodgroup = bloodgroup
    self.next = next
    self.prev = prev

class WRM:

  def __init__(self):
    self.dh = Patient(None,None,None,None,None,None)
    self.dh.next = self.dh
    self.dh.prev = self.dh

  def registerPatient(self,id, name, age, bloodgroup):

    new_patient = Patient(id,name,age,bloodgroup,None,None)
    last = self.dh.prev
    last.next = new_patient
    new_patient.prev = last
    new_patient.next = self.dh
```

```
    self.dh.prev = new_patient
    print("Patient registered successfulyy.")

  def servePatient(self):

    if self.dh.next == self.dh:
      print("No patients in the waiting room.")
      return
    first_patient = self.dh.next
    print(f"Serving patient: {first_patient.name}")
    self.dh.next = first_patient.next
    first_patient.next.prev = self.dh

  def showAllPatient(self):

    if self.dh.next == self.dh:
      print("No patients in the waiting room.")
      return
    current = self.dh.next
    print("Patients in the waiting room:")
    while current != self.dh:
      print(f"ID: {current.id}, Name: {current.name}, Age: {current.age}, Blood
Group: {current.bloodgroup}")
      current = current.next

  def canDoctorGoHome(self):

    if self.dh.next == self.dh:
      print("Yes, doctor can go home.")
      return True
    else:
      print("No, patients are still waiting")
      return False

  def cancelAll(self):

    self.dh.next = self.dh
    self.dh.prev = self.dh
    print("All appointments have been canceled")

  def ReverseTheLine(self):

    if self.dh.next == self.dh:
      print("No patients to reverse.")
      return
    current = self.dh
    while True:
      current.next = current.prev
      current.prev = current.next
      if current == self.dh:
```

```
        break
    print("Patient line reversed.")

def main():

  wrm = WRM()
  while True:
    print("\n**Welcome to Waiting Room Management System**")
    print("\nOptions:")
    print("1. Add Patient")
    print("2. Serve Patient")
    print("3. Show All Patient")
    print("4. Can Doctor Go Home?")
    print("5. Cancel All Appointments")
    print("6. Reverse The Line")
    print("7. Exit")

    choice = int(input("Enter your choice: "))

    if choice == 1:
      id = int(input("Enter patient ID: "))
      name = input("Enter patient name: ")
      age = int(input("Enter patient age: "))
      bloodgroup = input("Enter patient blood group: ")
      wrm.registerPatient(id,name,age,bloodgroup)
      print("Patient added successfully.")
    elif choice == 2:
      wrm.servePatient()
    elif choice == 3:
      wrm.showAllPatient()
    elif choice == 4:
      wrm.canDoctorGoHome()
    elif choice == 5:
      wrm.cancelAll()
    elif choice == 6:
      wrm.ReverseTheLine()
    elif choice == 7:
      print("Exiting the system. Goodbye!")
      break
    else:
      print("Invalid choice. Please try again.")
if __name__=="__main__":
  main()
```