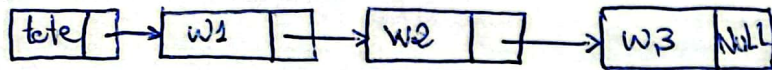


- Un train est composé d'une locomotive (tête) + wagons



- Un agent de maintenance doit, avant chaque départ, vérifier l'état du train.
- Un train part avec n wagons selon nombre des voyageurs
- L'agent ne peut pas accéder au train que par la locomotive
- S'il y a un pb au niveau de la locomotive, il demande d'ajouter une nouvelle locomotive en tête sans supprimer celle qui est panne
- L'agent peut ajouter nouveau wagon à la fin du train selon nombre des voyageurs

Partie 1: 1 - Définir structure "wagon" caractérisée par

Numéro

type : locomotive ou wagon

état : en panne ou OK

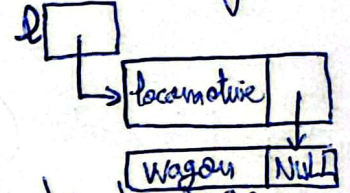
- 2 - Créer la liste chaînée représentée du train
- 3 - ajouter locomotive en tête du train
- 4 - ajouter wagon à la queue du train.
- 5 - Chercher wagon selon numéro, si elle existe
→ sinon 0
- 6 - afficher train

①

1. type de struct wagon

```

{
    int num;
    char T; // l si locomotive, w si wagon
    int etat;
    struct wagon* suiv;
} wagon;
  
```



2. Soit N: nbr des voyageurs, nb: nbr de wagon.

```

struct wagon* l; aux;
l = (struct wagon*) malloc(sizeof(struct wagon));
  
```

l → num = 1;

l → T = 'l';

l → etat = 1;

x = N % 33;

if (x)

nb = (N/33) + 1;

else

nb = N/33

aux = l;

while (nb) // (while nb > 0) {

```

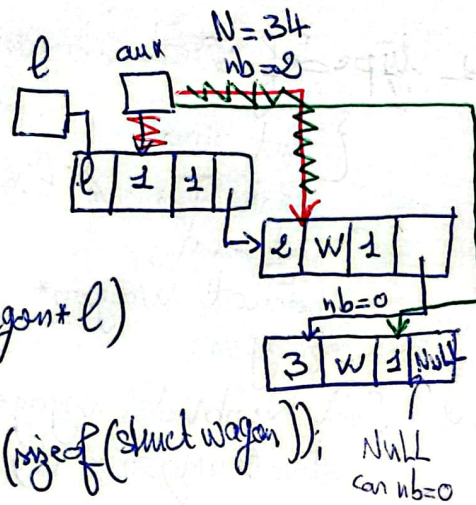
    aux → suiv = (struct wagon*) malloc(sizeof(struct wagon));
    (aux → suiv) → num = x;
    x++;
    (aux → suiv) → T = 'w';
    (aux → suiv) → etat = 1;
  }
  
```

②


```

nb--;
aux = aux->suiv;
}
aux->suiv = NULL;
}

```



3 - void ajout_locomotive(struct wagon * l)

```

{ struct wagon * new;
new = (struct wagon *) malloc(sizeof(struct wagon));
new->num = 1;
new->T = 'l';
new->etat = 1;
}

```

un agent de maintenance doit, avant ~~que~~ chaque depart, verifier l'etat du train.

```

l->etat = 0;
new->suiv = l;
l->new;
}

```

4 - void ajout_wagon(struct wagon * l)

```

{ struct wagon * aux = l;
while (aux->suiv)
aux = aux->suiv;
aux->suiv = (struct wagon *) malloc(sizeof(struct wagon));
(aux->suiv)->num = (aux->num) + 1;
(aux->suiv)->T = 'w';
(aux->suiv)->etat = 1;
(aux->suiv)->suiv = NULL;
}

```

(3)

(4)