

```

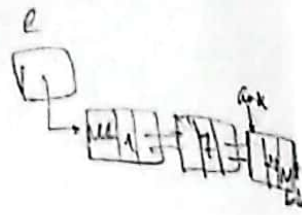
1) struct Cellule
{ int v;
  struct Cellule* suiv, prec; } cellule;
cellule* liste;

```

```

2) int debut (int a, liste l)
{ if (l != NULL)
  { if (l->v == a)
    { return 1; }
    return 0; }
}

```



```

int fin (int a, liste l)
{ liste aux = l;
  if (l != NULL)
  { while (aux->suiv != NULL)
    { aux = aux->suiv;
      if (aux->v == a)
        return 1; }
    return 0; }
}

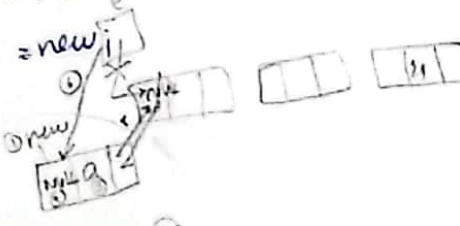
```

liste ajout-tête (int a, liste l)

```

1) liste new;
2) new = (cellule*) malloc (sizeof (cellule));
3) new->v = a;
4) new->prec = NULL;
5) new->suiv = l;
6) l->prec = new;
7) l = new;
return l;

```



liste ajout-fin (int a, liste l)

```

{ liste new; aux = l;
  new = (cellule*) malloc (sizeof (cellule));
  new->v = a;
  while (aux->suiv != NULL)
    aux = aux->suiv;
  new->prec = aux;
  new->suiv = NULL;
  aux->suiv = new;
}

```



```

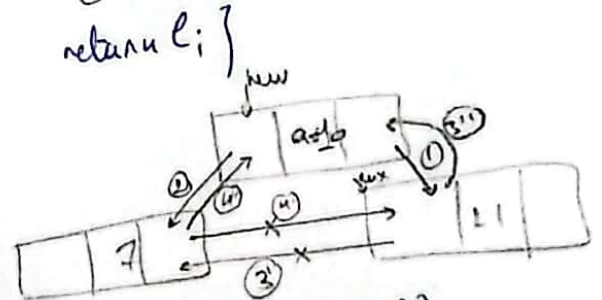
4) void afficher (liste aux)
{ while (aux != NULL)
  { printf ("%d ", aux->v);
    aux = aux->suiv; }
}

```

```

5) liste ajout (int a, liste l)
{ liste aux = l, new;
  while (aux->v < a & aux->suiv != NULL)
    aux = aux->suiv;
  if (aux->suiv == NULL)
    return (ajout-tête (a, l));
  else if (aux->prec == NULL)
    return (ajout-tête (a, l));
  else {
    new = (cellule*) malloc (sizeof (cellule));
    new->v = a;
    1) new->suiv = aux;
    2) new->prec = aux->prec;
    3) (aux->prec)->suiv = new;
    4) aux->prec = new;
  }
  return l;
}

```



6) liste supprimer (int a, liste l)

```

{ liste aux = l;
  while (aux->v != a & aux != NULL)
    aux = aux->suiv;
  if (aux->v == a)
  { if (aux->suiv == NULL) // dernière cellule
    { (aux->prec)->suiv = NULL;

```

```

    } free(aux);
    return l; }

if (aux->prec == NULL)
{
    (aux->next)->prec = NULL;
    l = aux->next;
    free(aux);
    return l; } // au milieu

(aux->prec)->next = aux->next;
(aux->next)->prec = aux->prec;
free(aux);
return l; }

```