Linked List – insert_after

Team Emertxe

Linked List

Data Structure – Linked List

Analysis – insert_after

Analysis: Logic /Cases

Flowchart

Algorithm

Code

Linked List – Insert after - Analysis

Analysis – insert_after

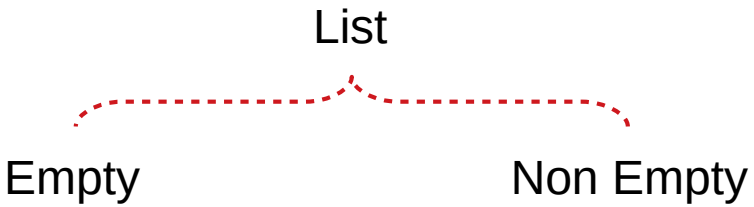
Analysis – insert_after

Cases :

List

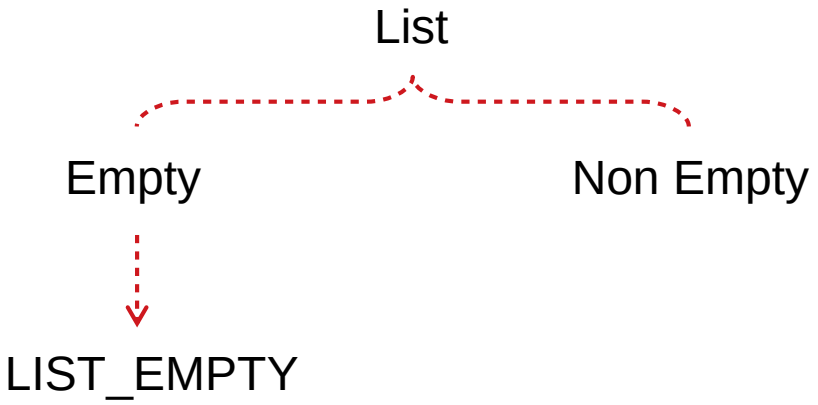
Analysis – insert_after

Cases :



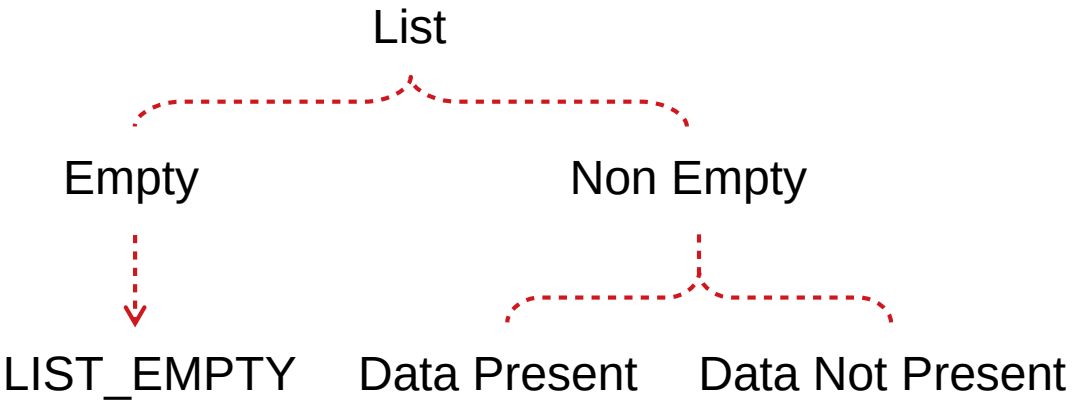
Analysis – insert_after

Cases :



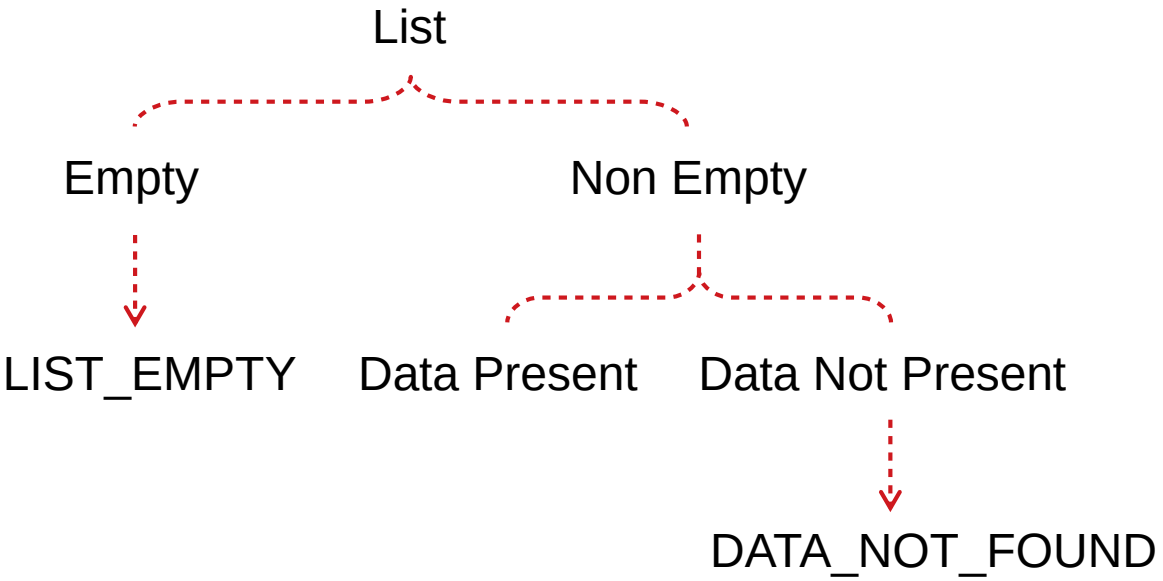
Analysis – insert_after

Cases :

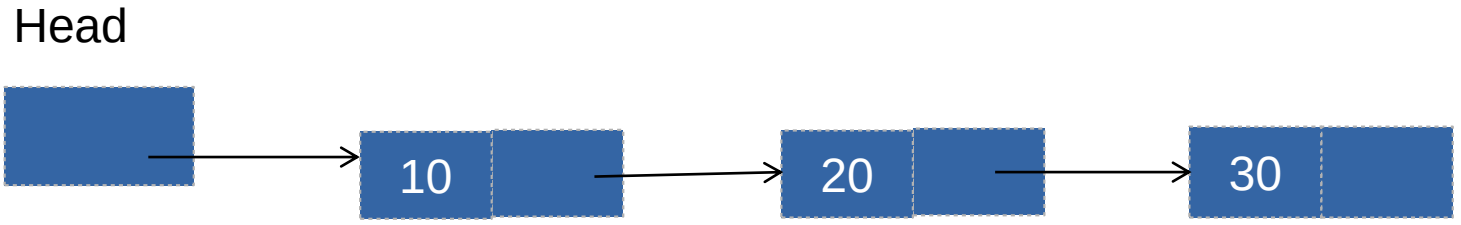


Analysis – insert_after

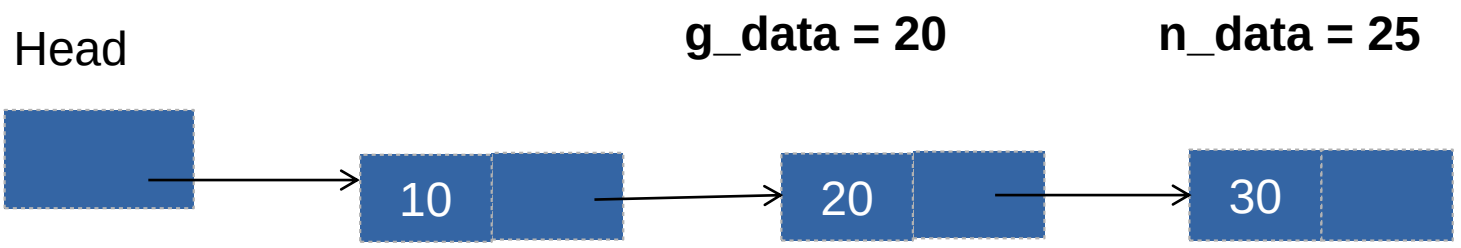
Cases :



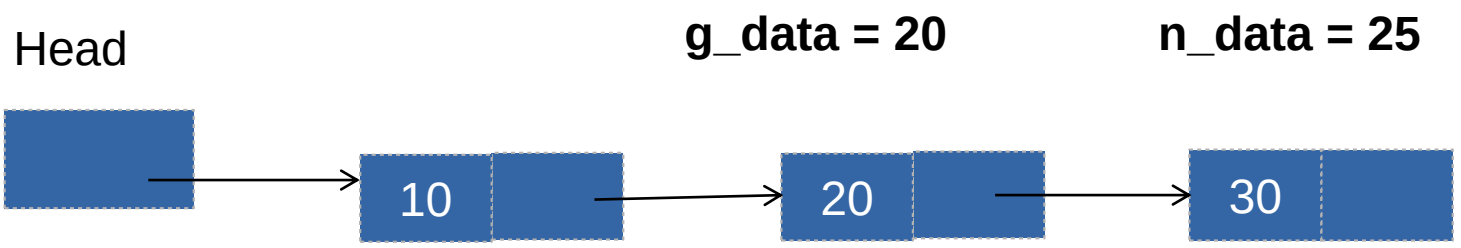
Analysis – insert_after



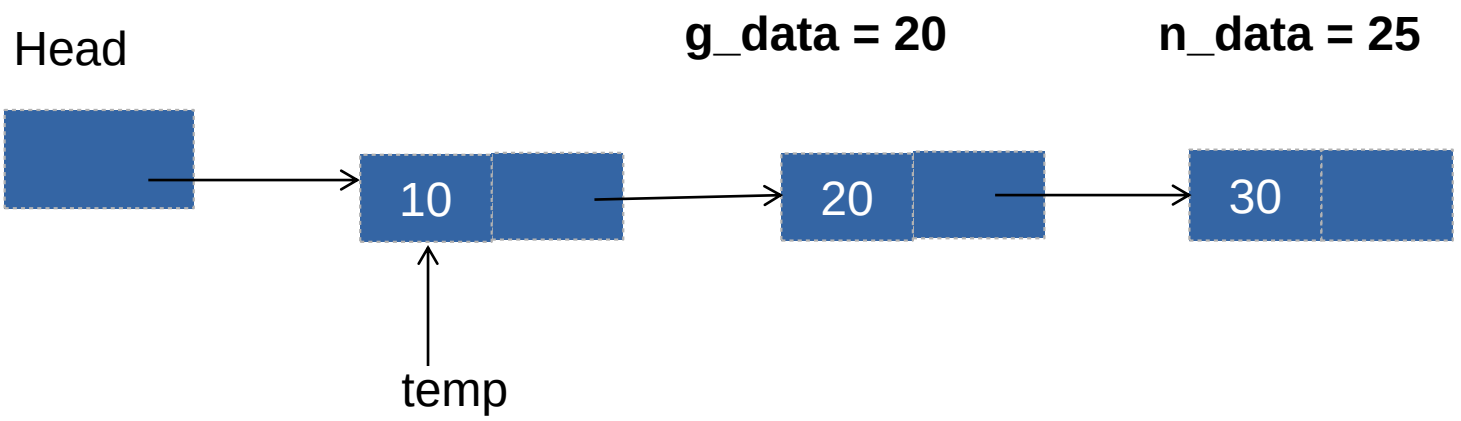
Analysis – insert_after



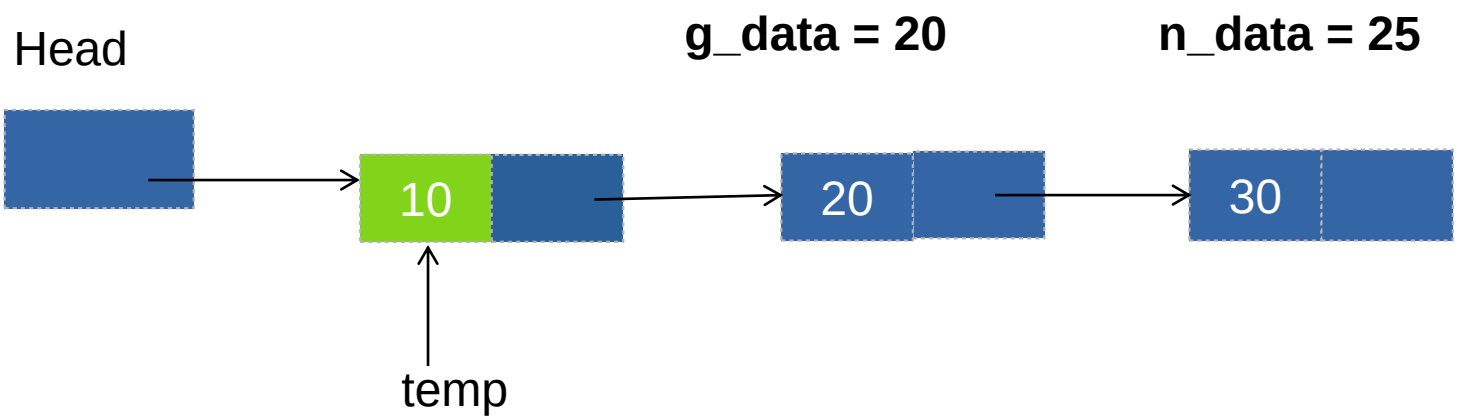
Analysis – insert_after



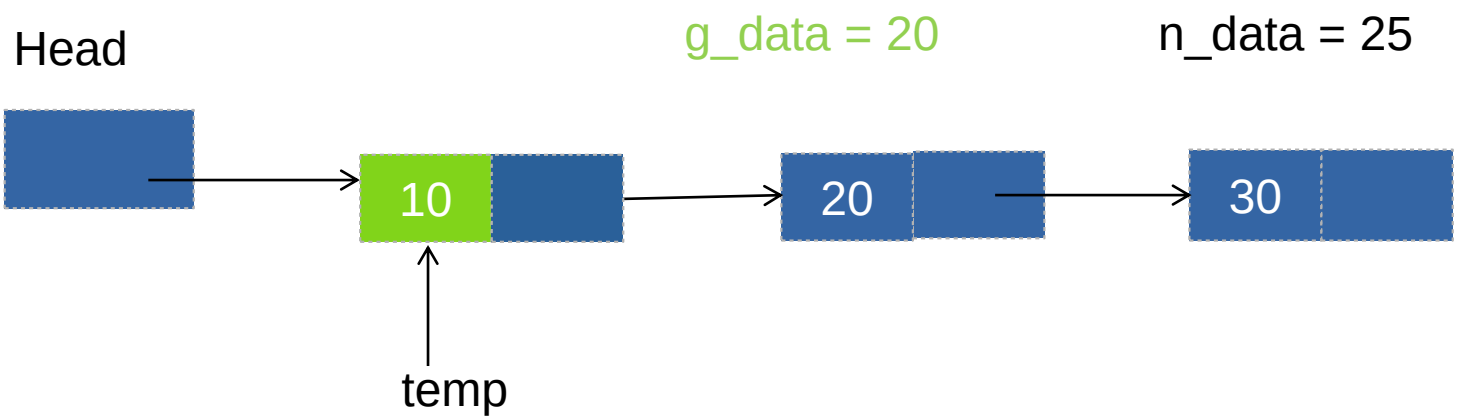
Analysis – insert_after



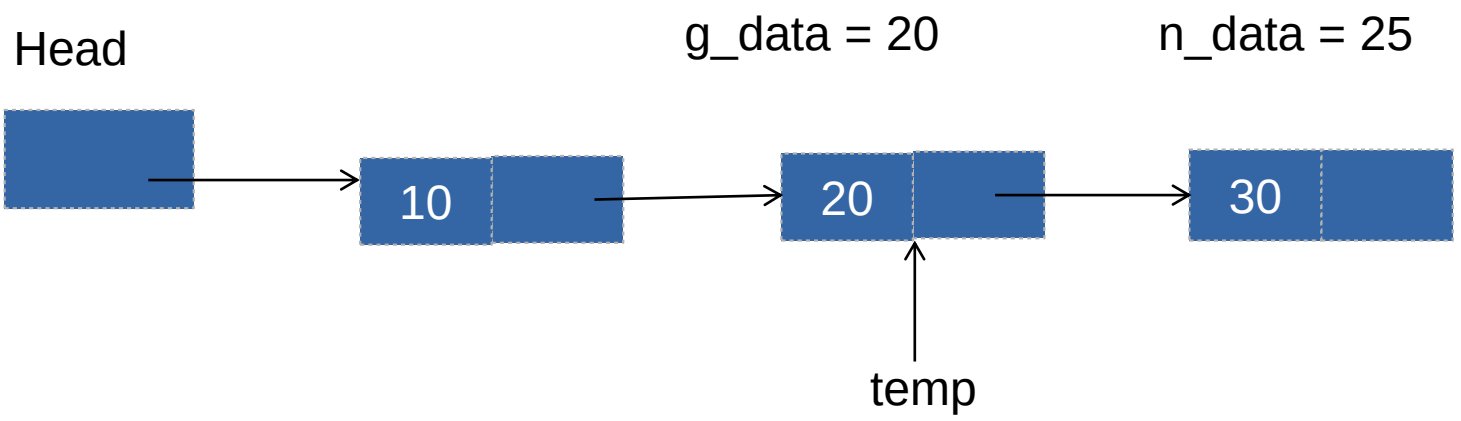
Analysis – insert_after



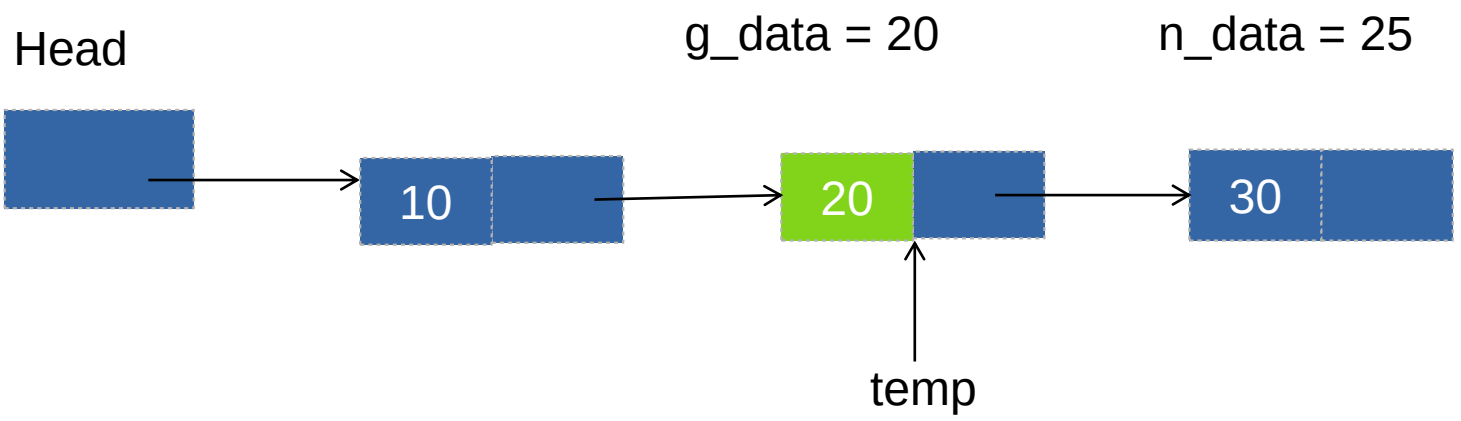
Analysis – insert_after



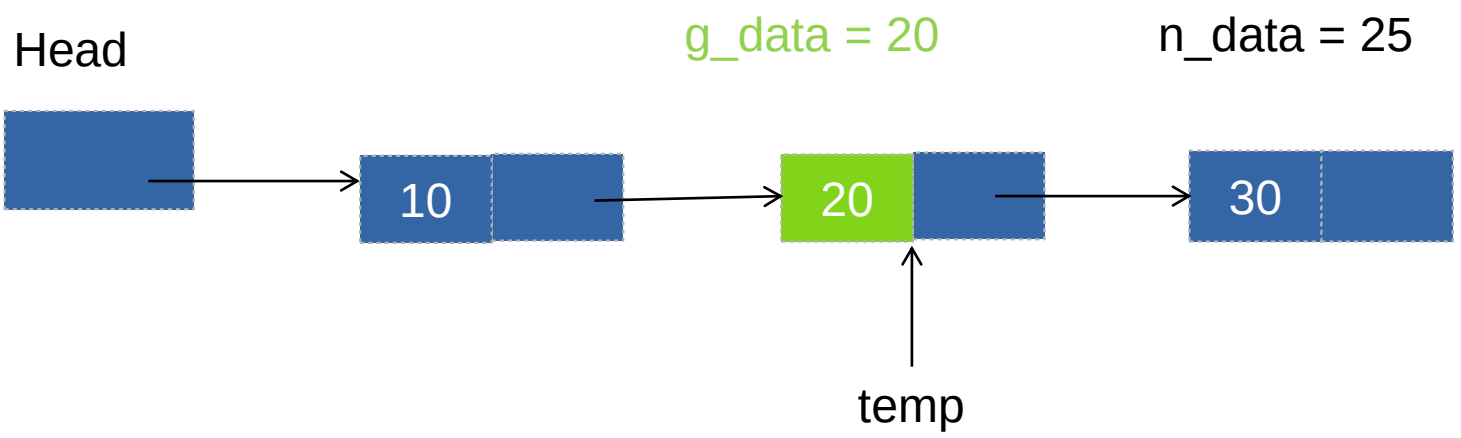
Analysis – insert_after



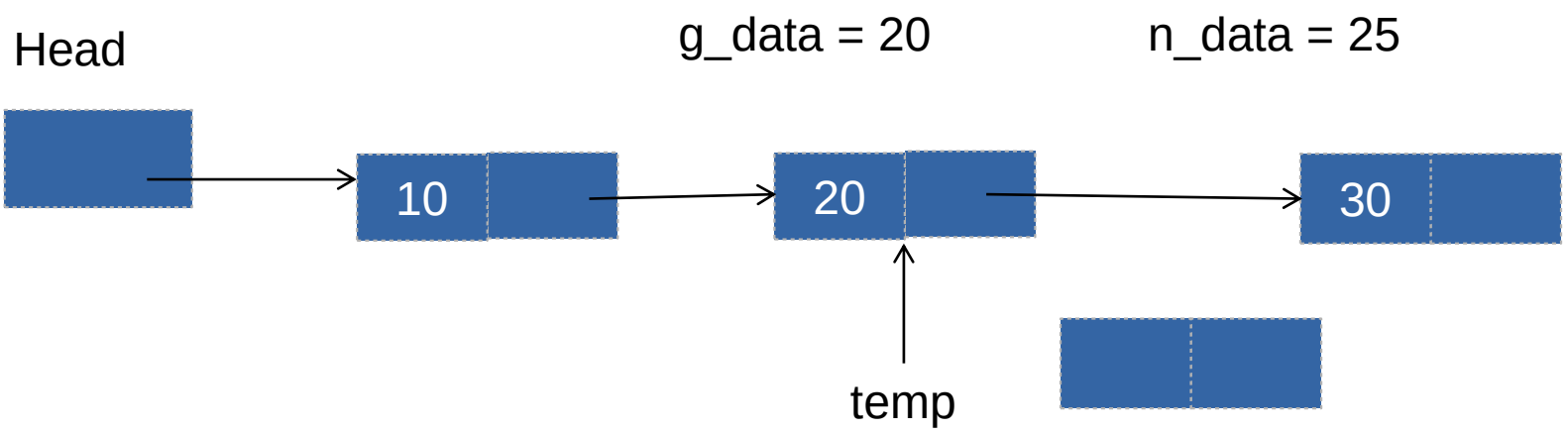
Analysis – insert_after



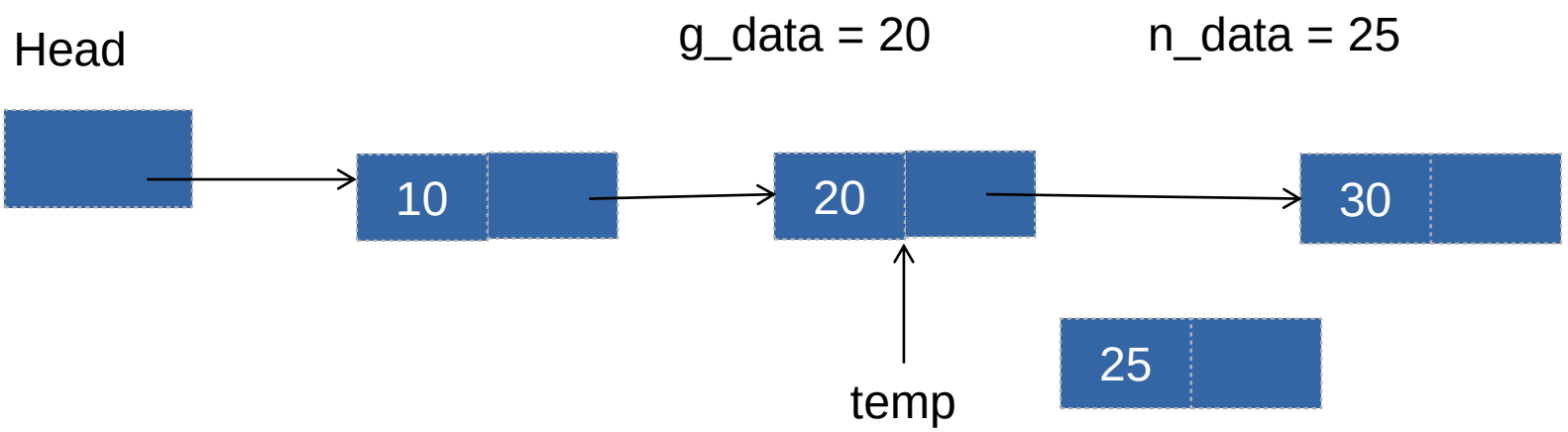
Analysis – insert_after



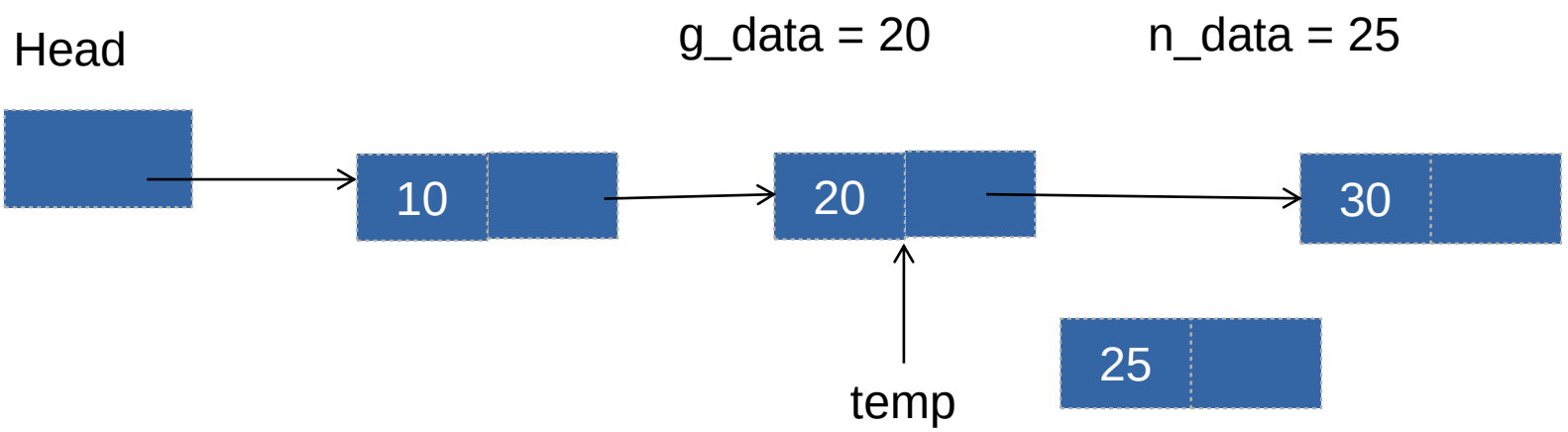
Analysis – insert_after



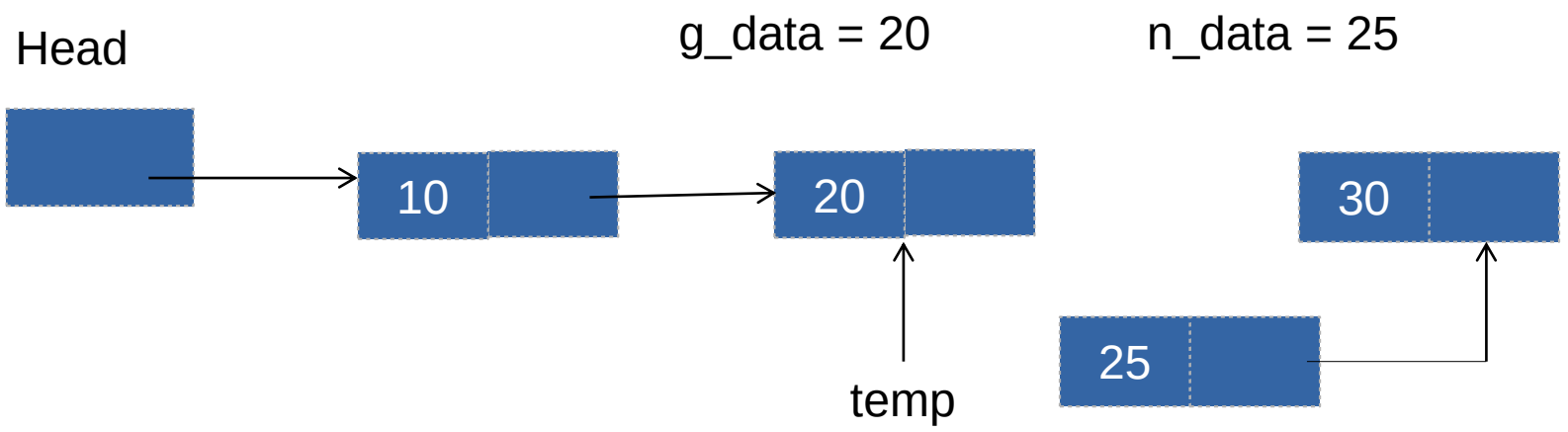
Analysis – insert_after



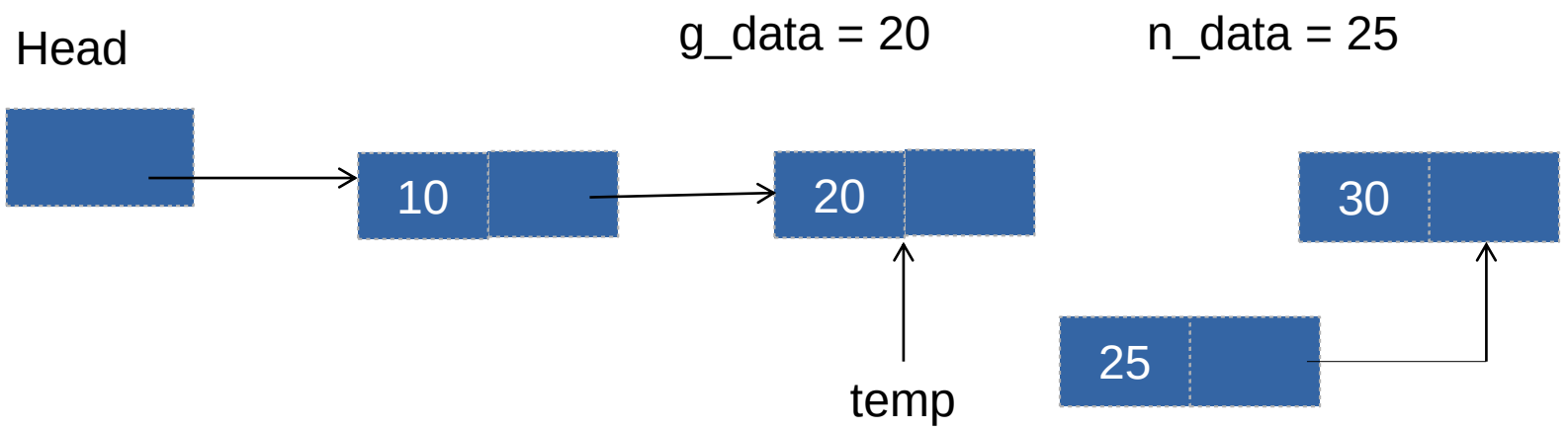
Analysis – insert_after



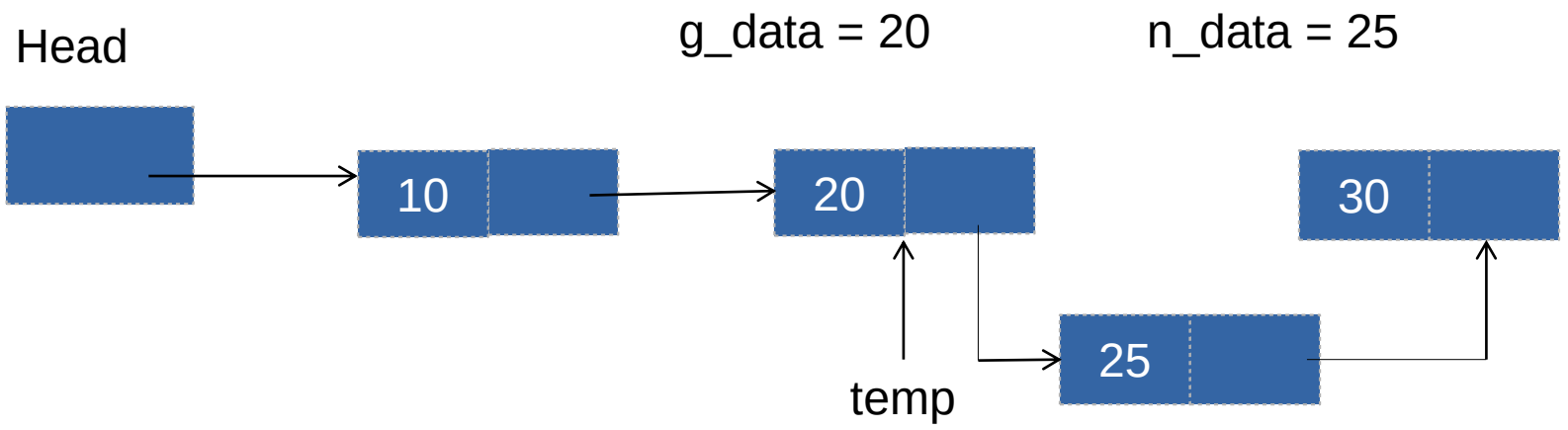
Analysis – insert_after



Analysis – insert_after



Analysis – insert_after



Linked List – Insert after - Flowchart

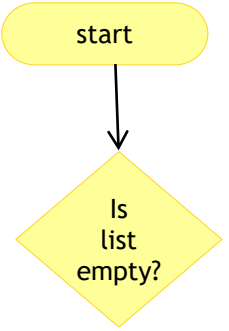
Flowchart

insert_after

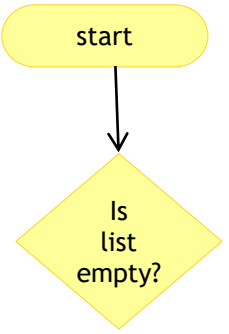
start

Flowchart
insert_after

Flowchart
insert_after

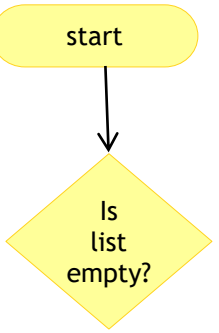


Flowchart
insert_after



Head

Flowchart
insert_after

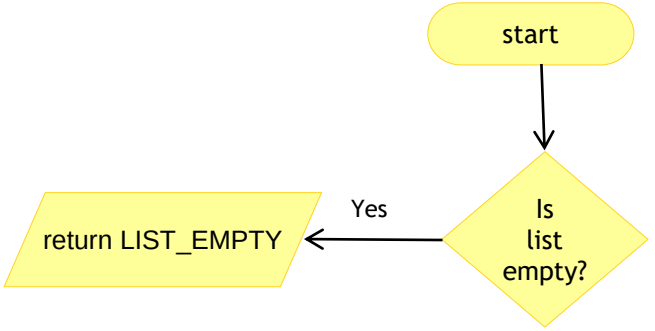


Head

NULL

EMERTXE

Flowchart
insert_after



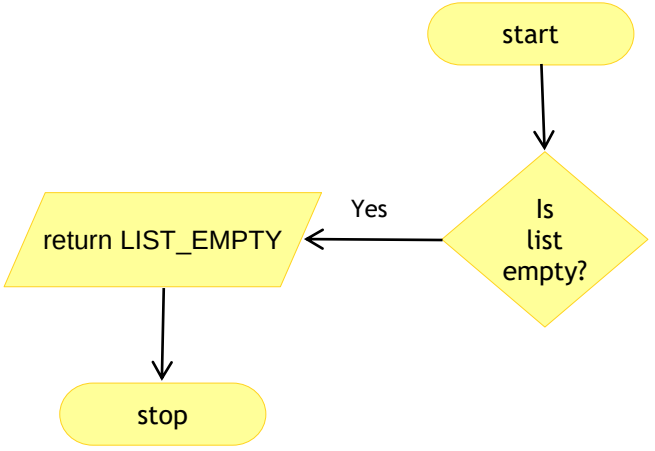
Head

NULL

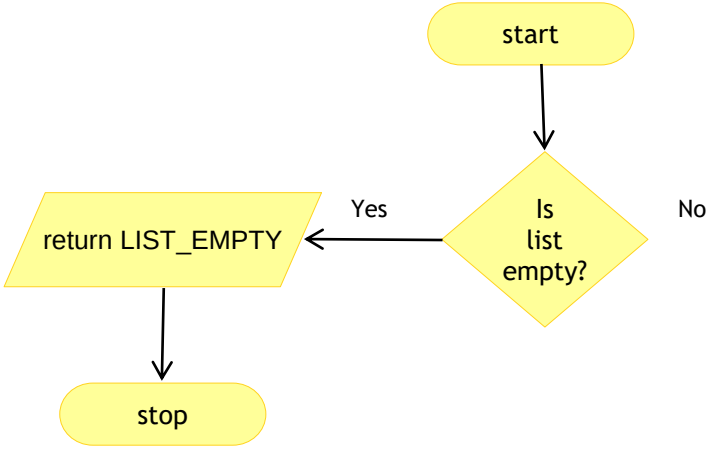
Flowchart
insert_after

Head

NULL



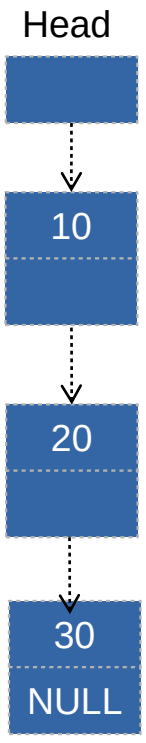
Flowchart
insert_after

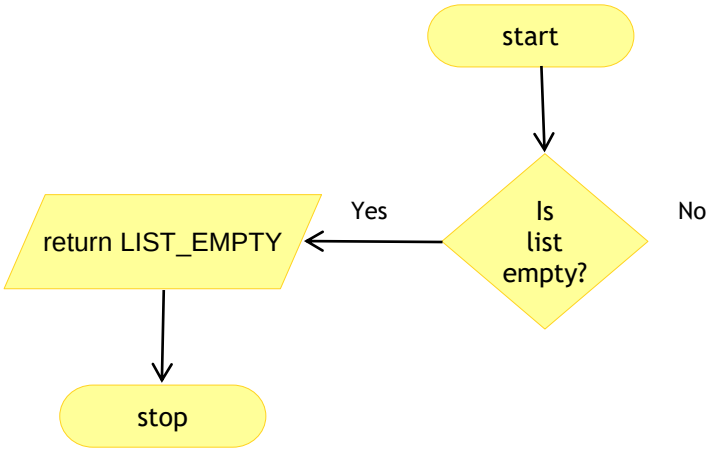


Head

NULL

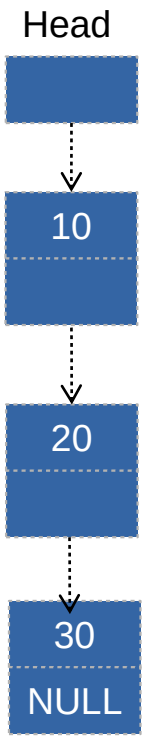
Flowchart
insert_after





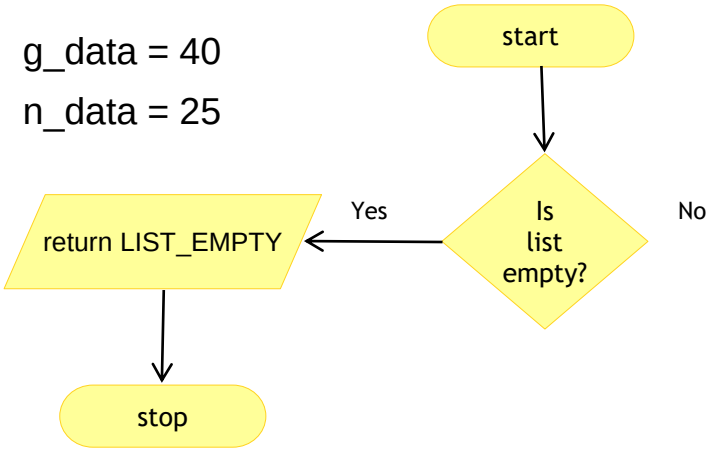
EMERTXE

Flowchart
insert_after

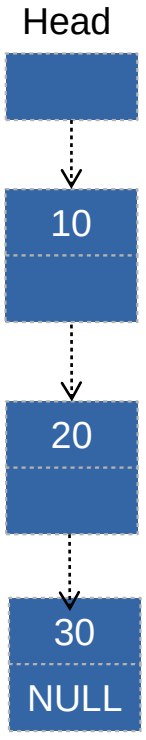


g_data = 40

n_data = 25

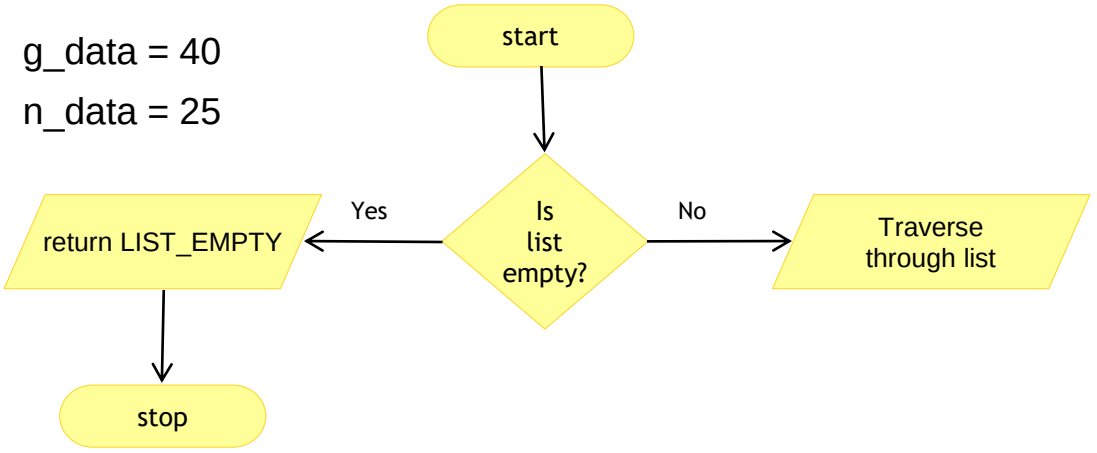


Flowchart
insert_after

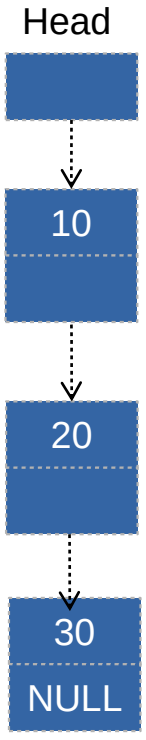


g_data = 40

n_data = 25

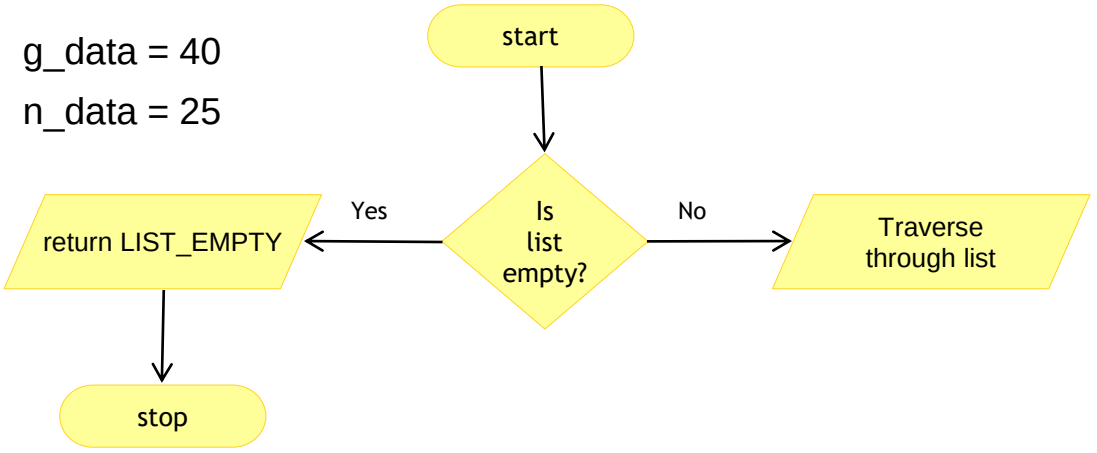


Flowchart
insert_after

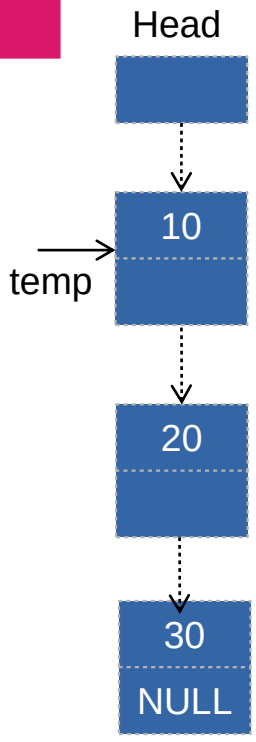


g_data = 40

n_data = 25

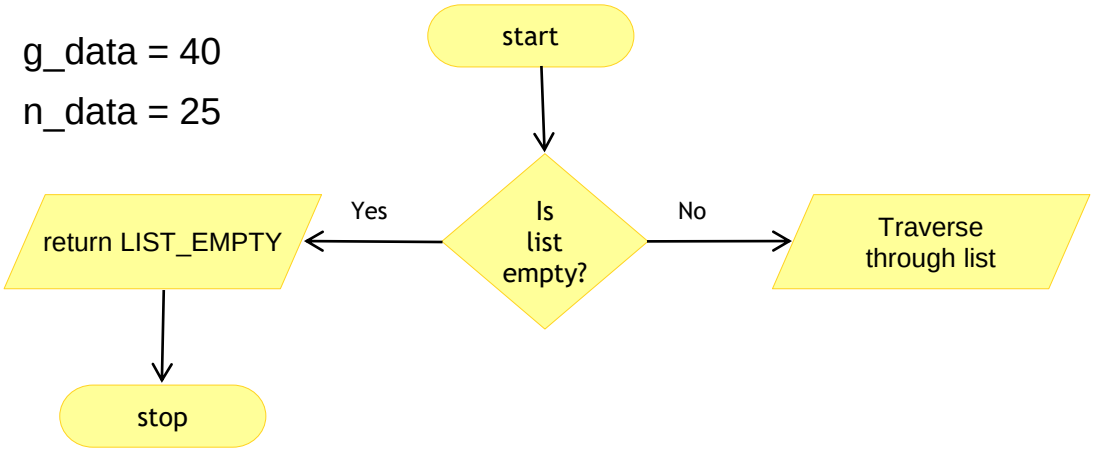


Flowchart
insert_after

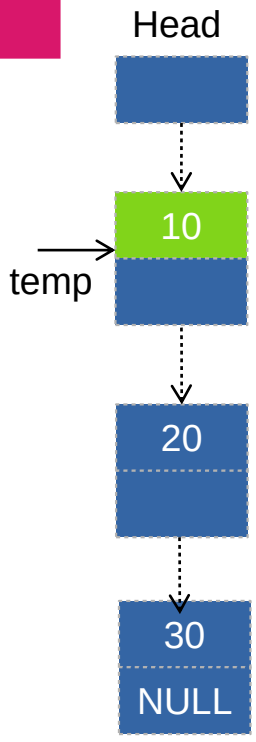


g_data = 40

n_data = 25

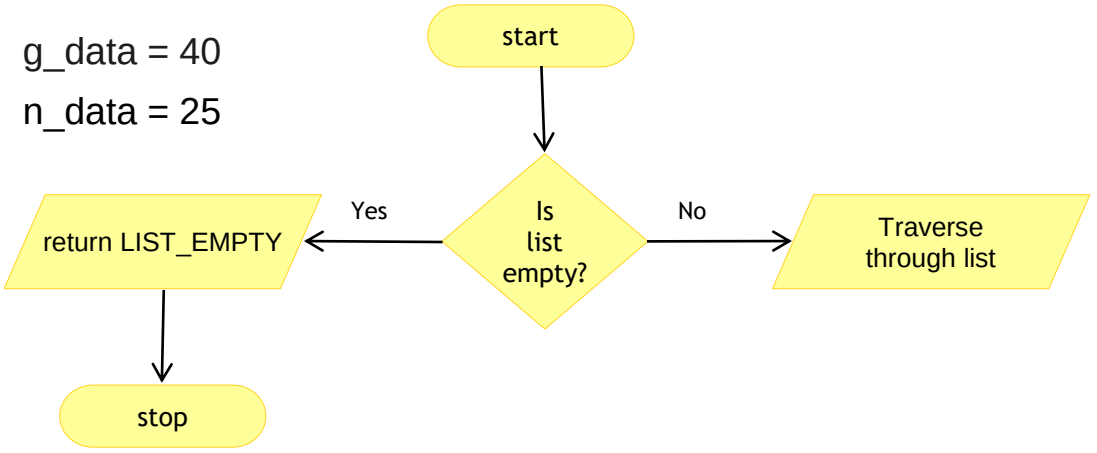


Flowchart
insert_after

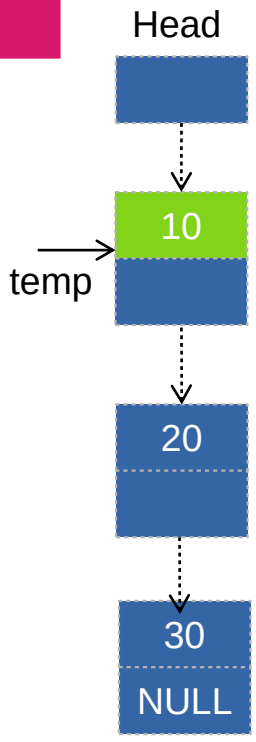


g_data = 40

n_data = 25

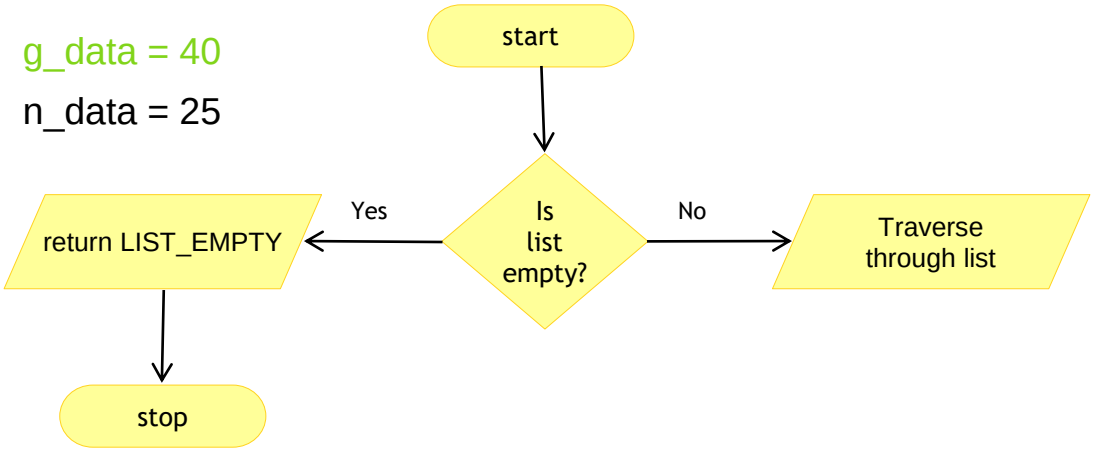


Flowchart
insert_after

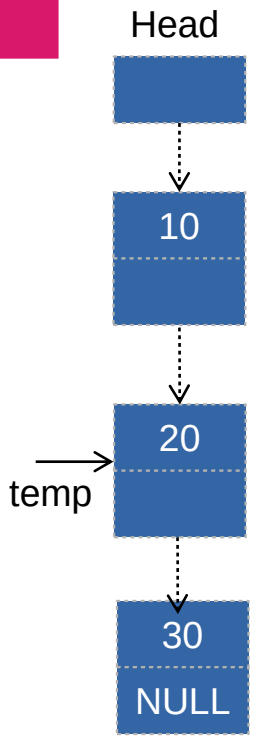


g_data = 40

n_data = 25

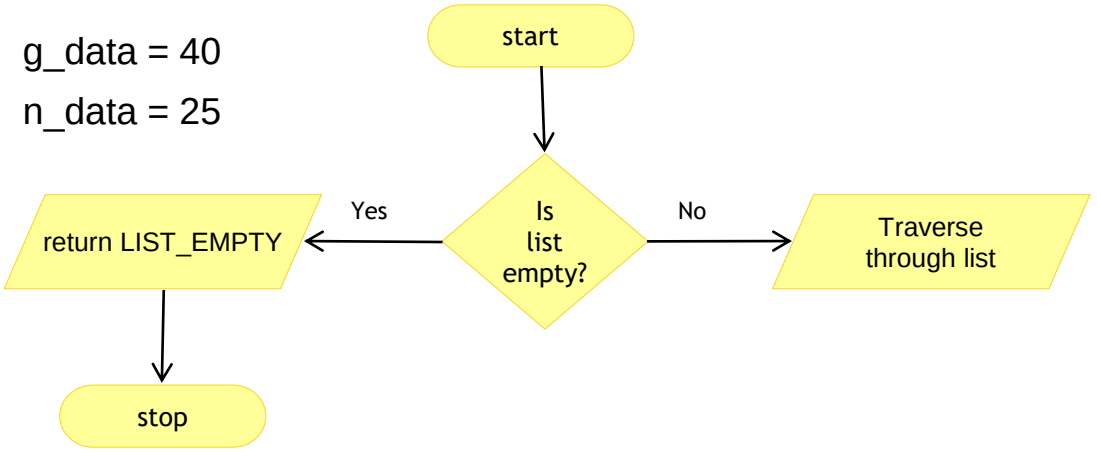


Flowchart
insert_after

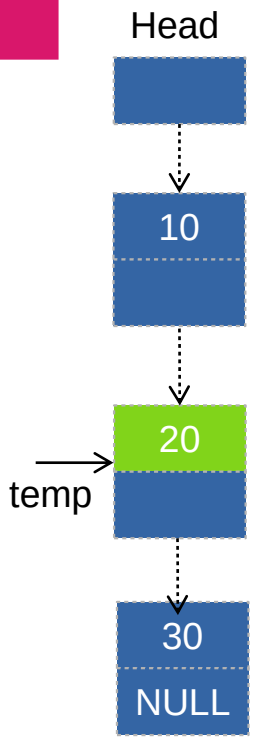


g_data = 40

n_data = 25

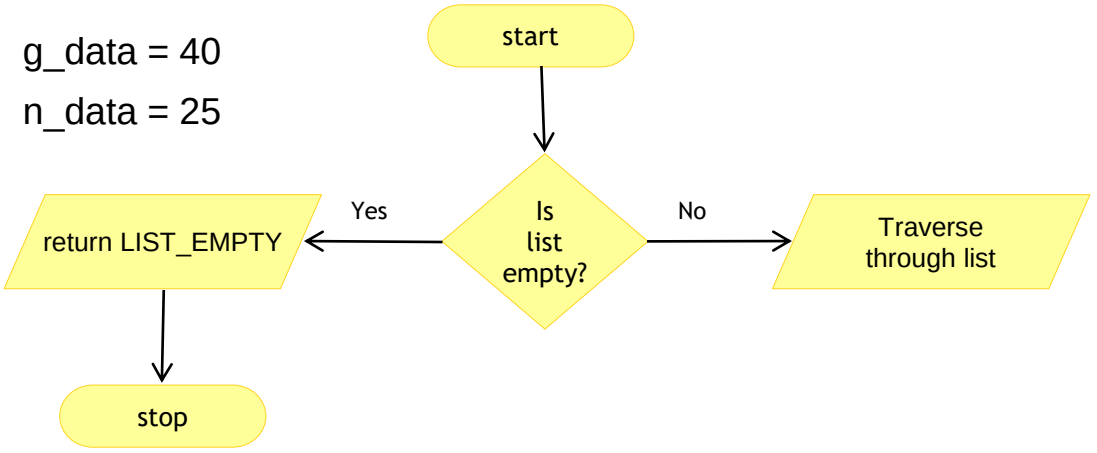


Flowchart
insert_after

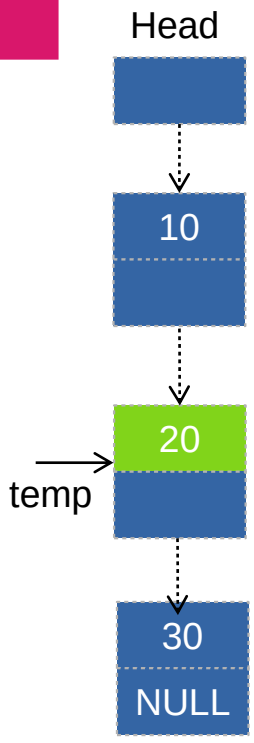


g_data = 40

n_data = 25

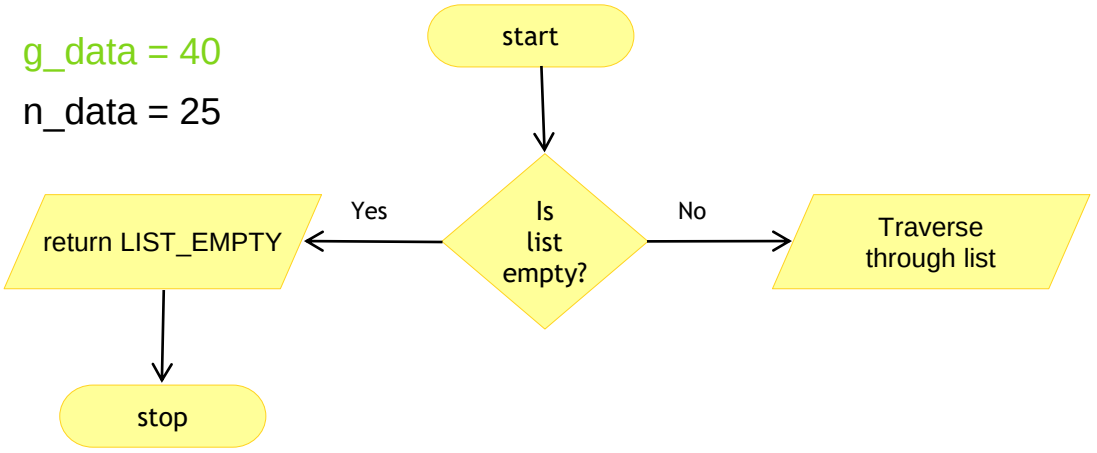


Flowchart
insert_after

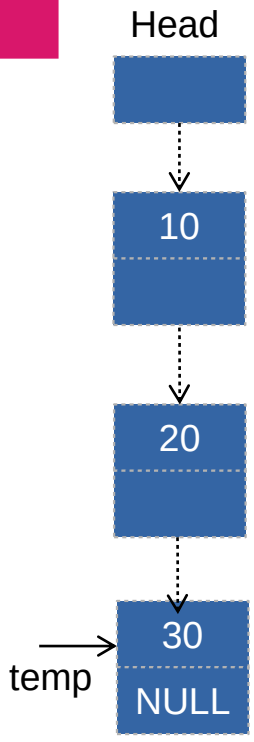


g_data = 40

n_data = 25

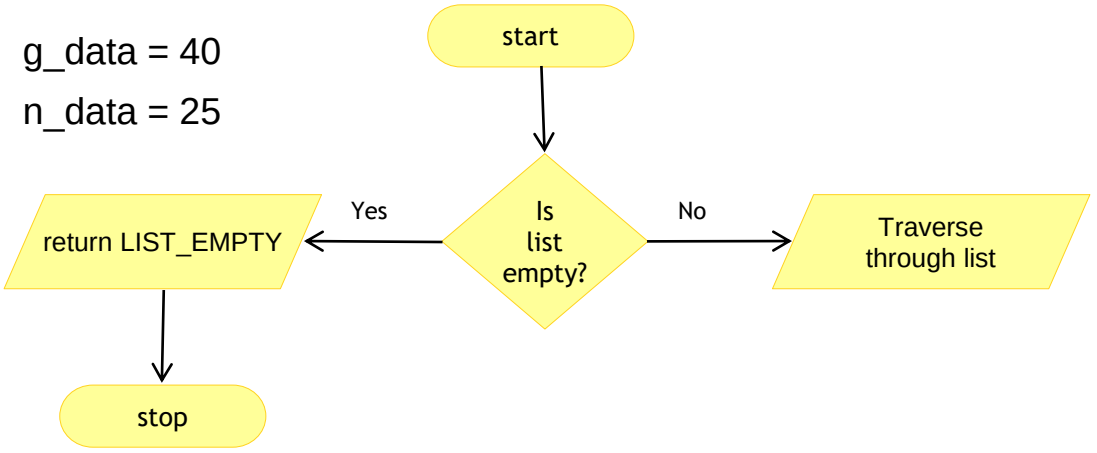


Flowchart
insert_after

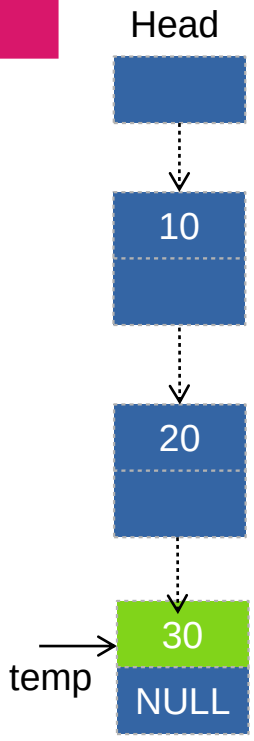


g_data = 40

n_data = 25

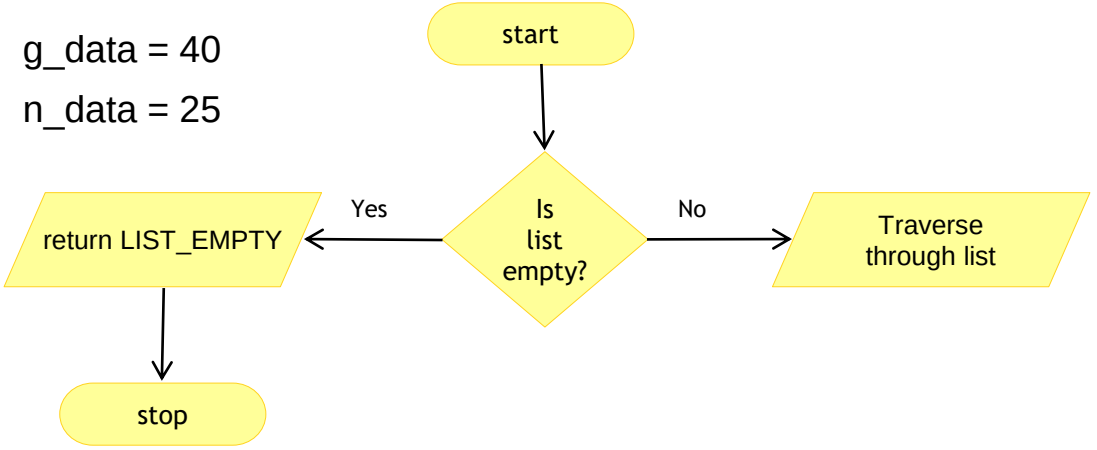


Flowchart
insert_after

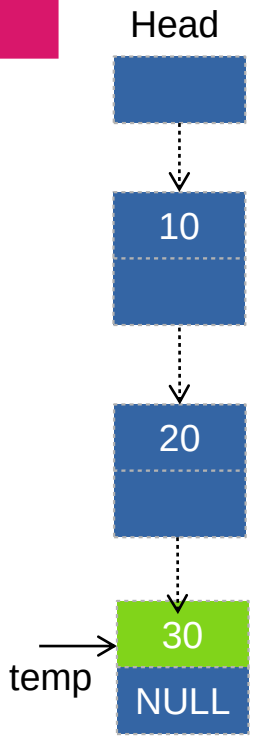


g_data = 40

n_data = 25

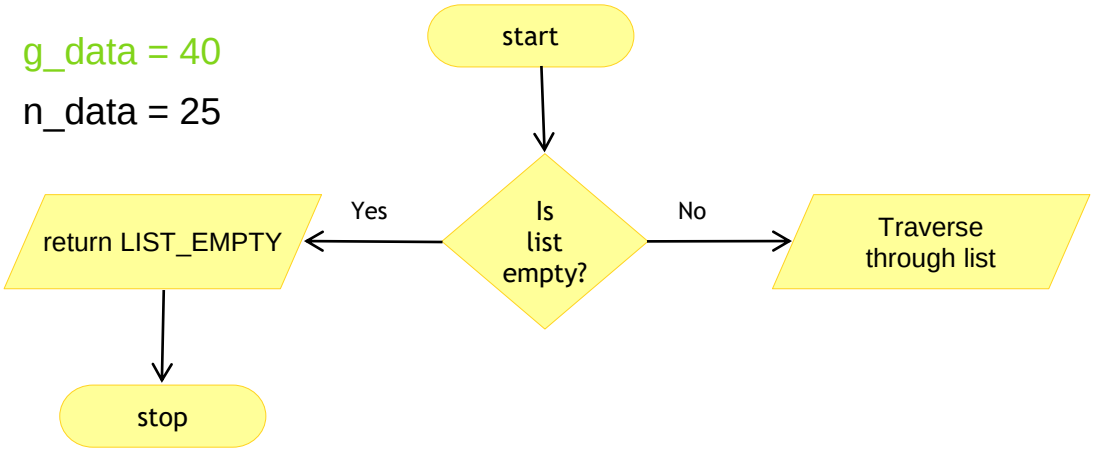


Flowchart
insert_after

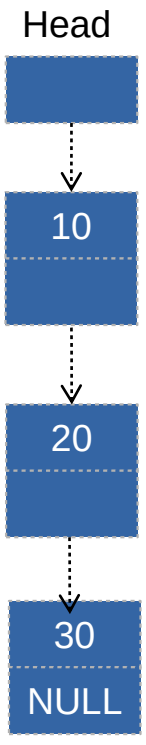


g_data = 40

n_data = 25

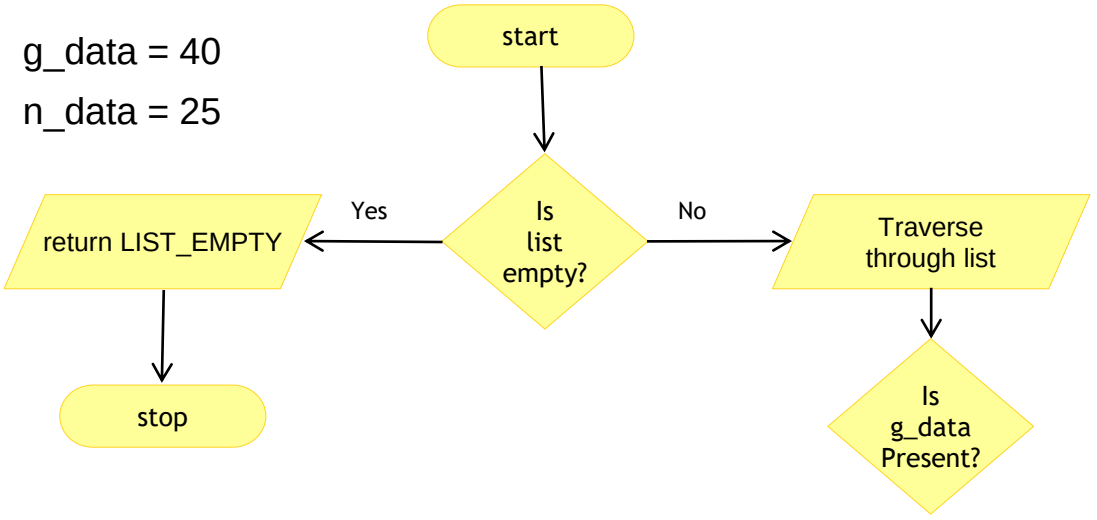


Flowchart
insert_after

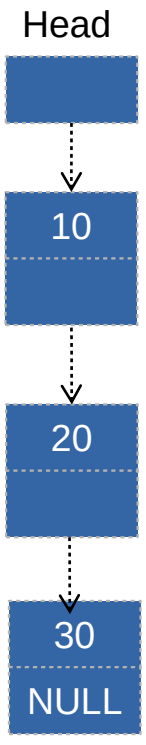


g_data = 40

n_data = 25

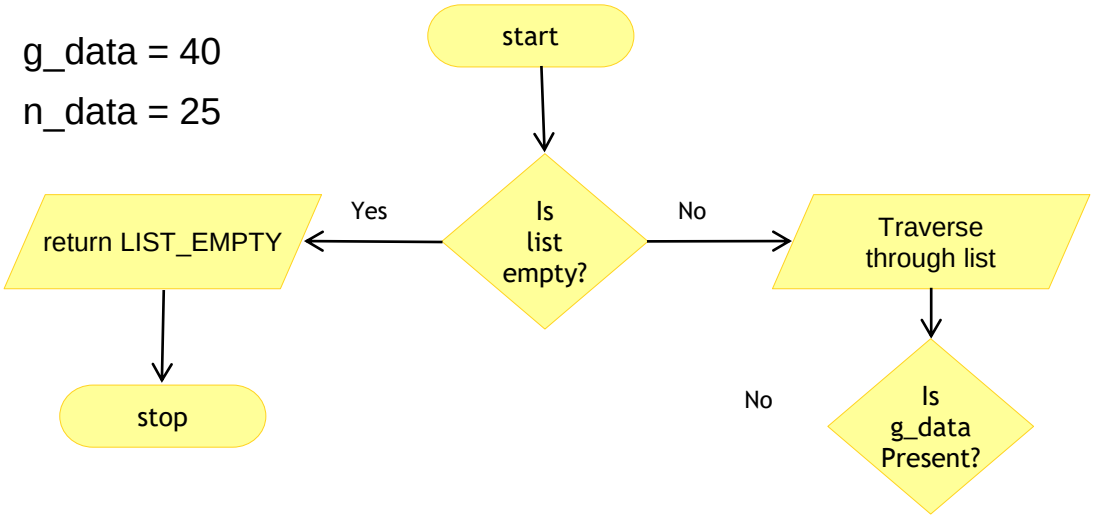


Flowchart
insert_after



g_data = 40

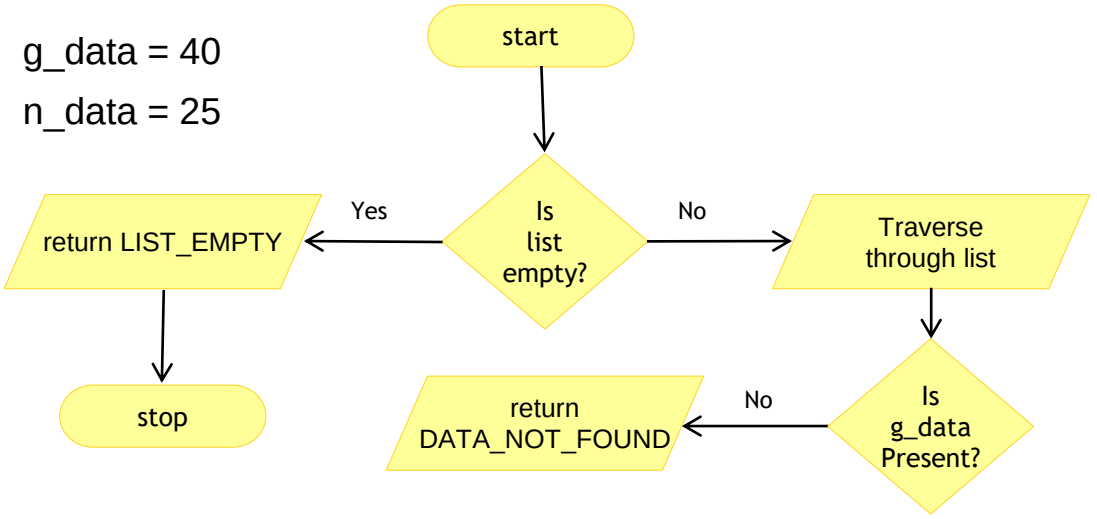
n_data = 25



Flowchart
insert_after

g_data = 40

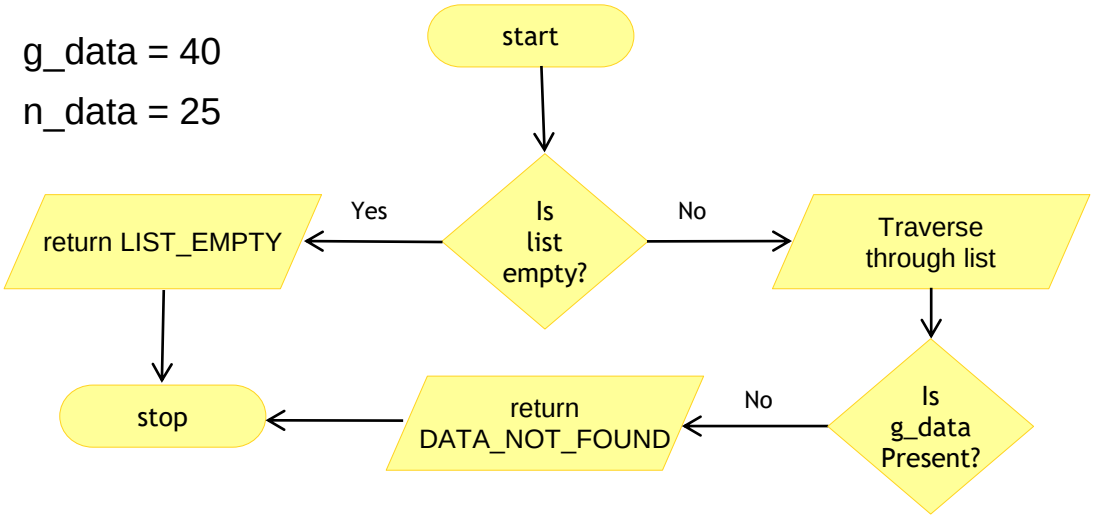
n_data = 25



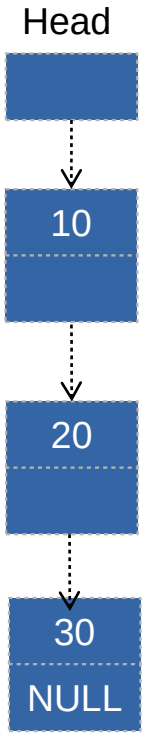
Flowchart
insert_after

g_data = 40

n_data = 25

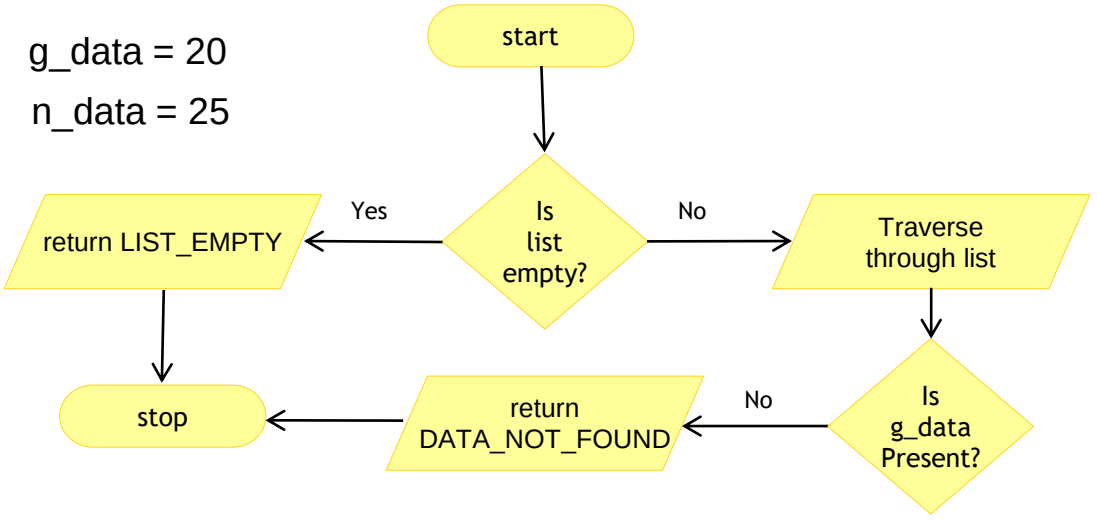


Flowchart
insert_after

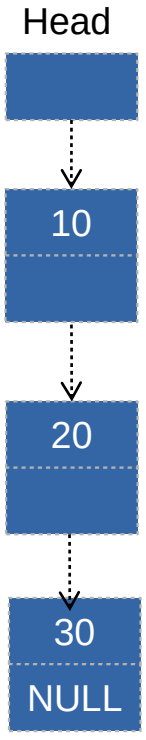


g_data = 20

n_data = 25

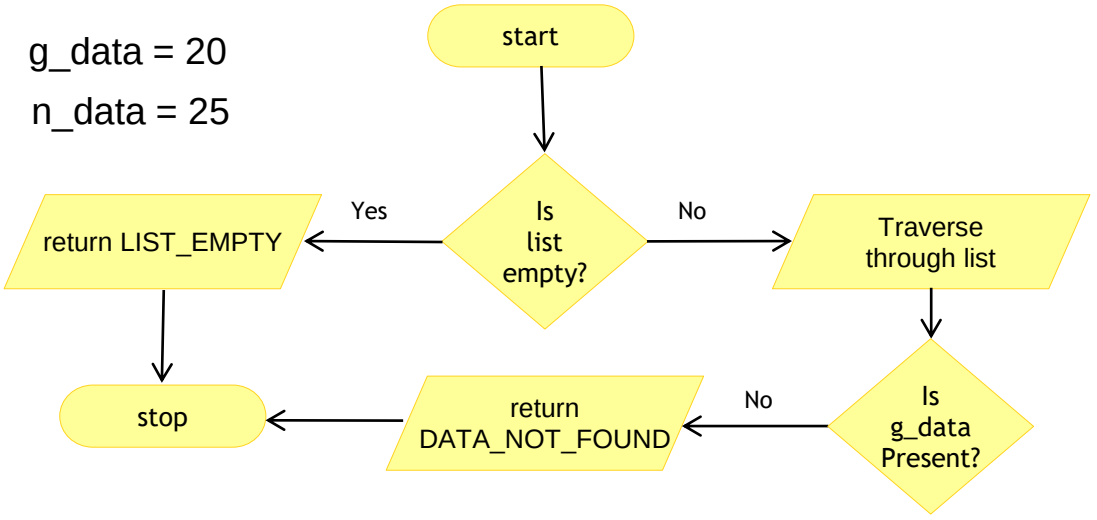


Flowchart
insert_after

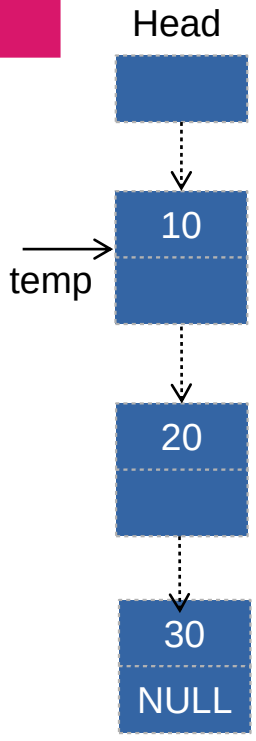


g_data = 20

n_data = 25

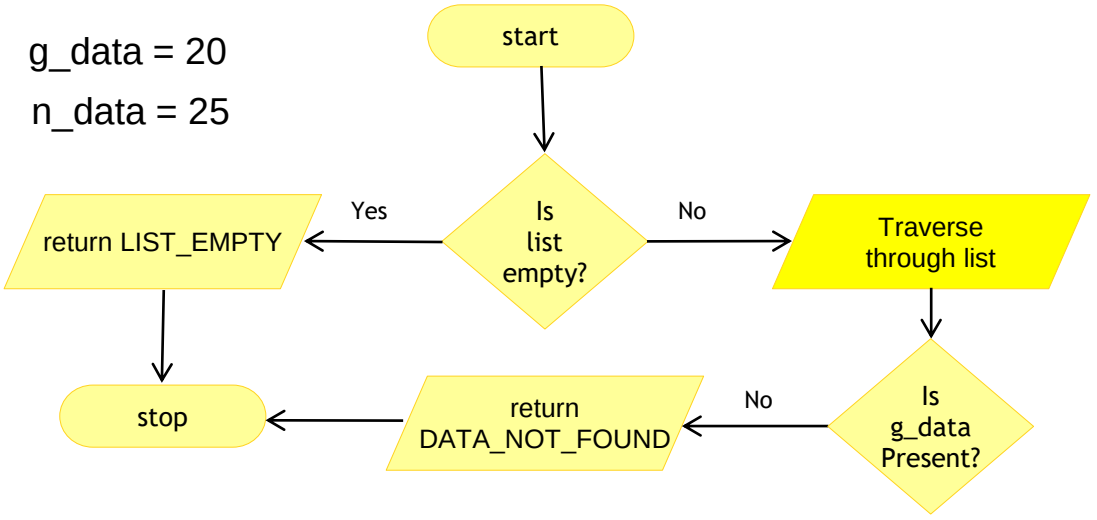


Flowchart
insert_after

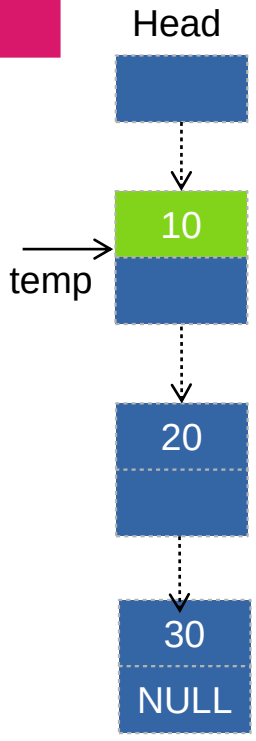


g_data = 20

n_data = 25

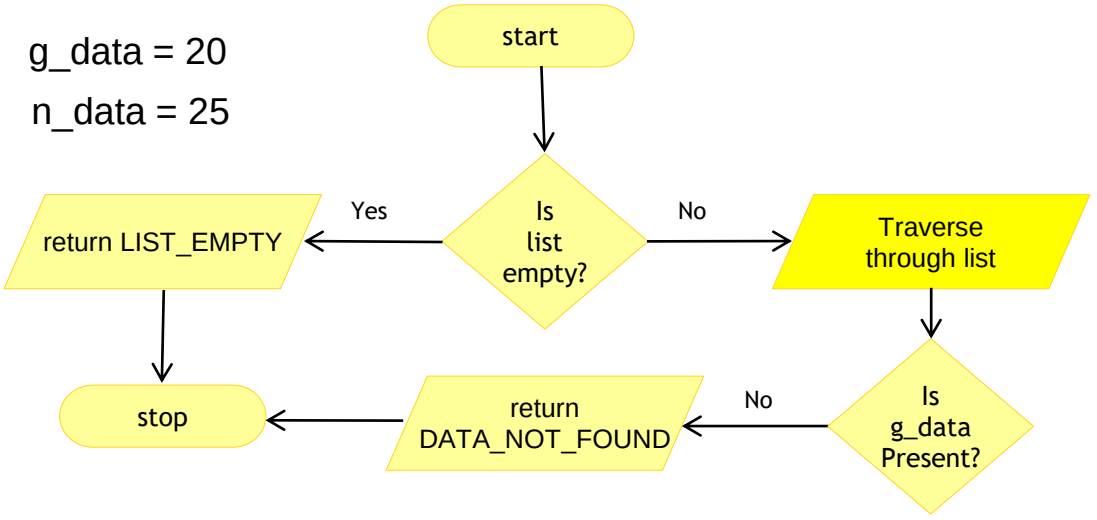


Flowchart
insert_after

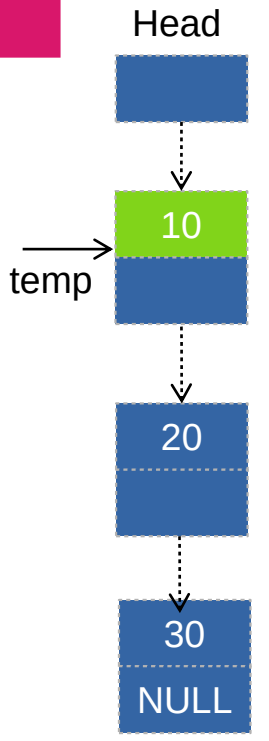


g_data = 20

n_data = 25

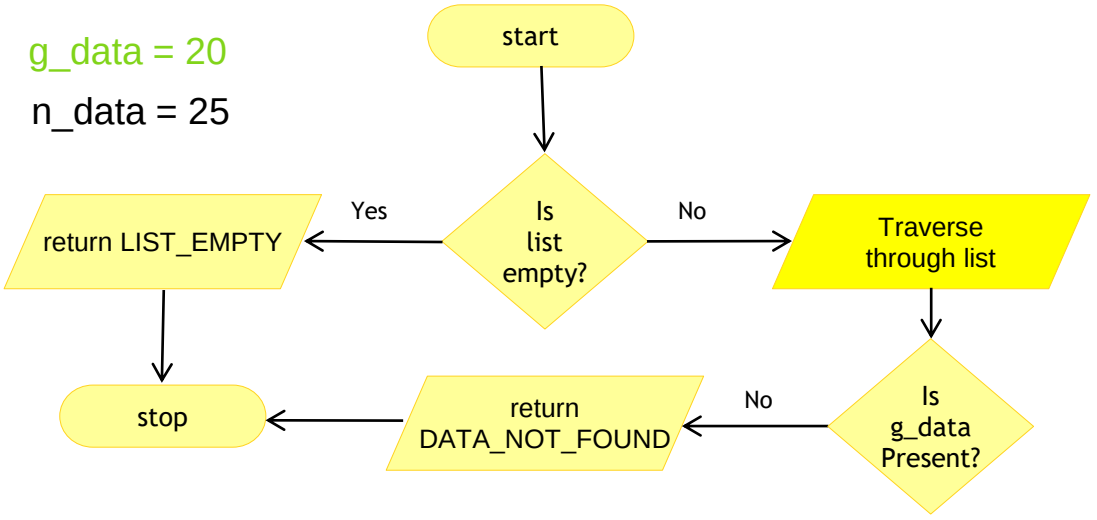


Flowchart
insert_after

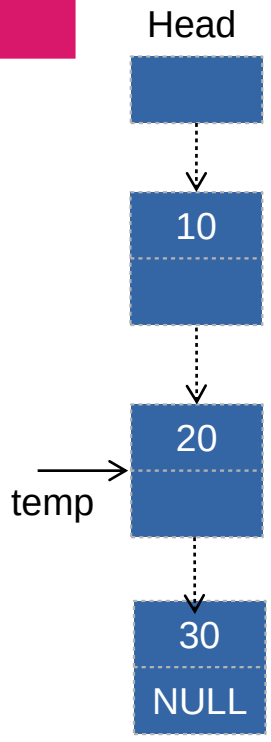


g_data = 20

n_data = 25

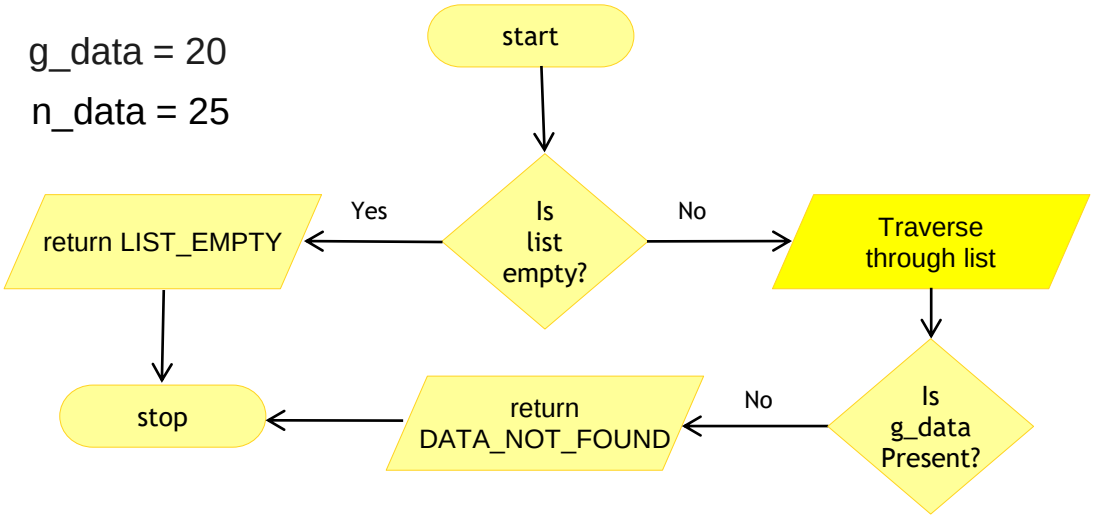


Flowchart
insert_after

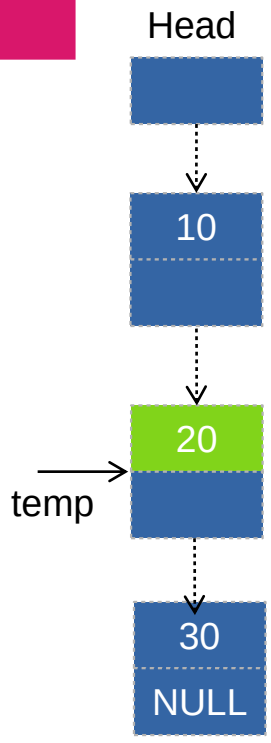


g_data = 20

n_data = 25

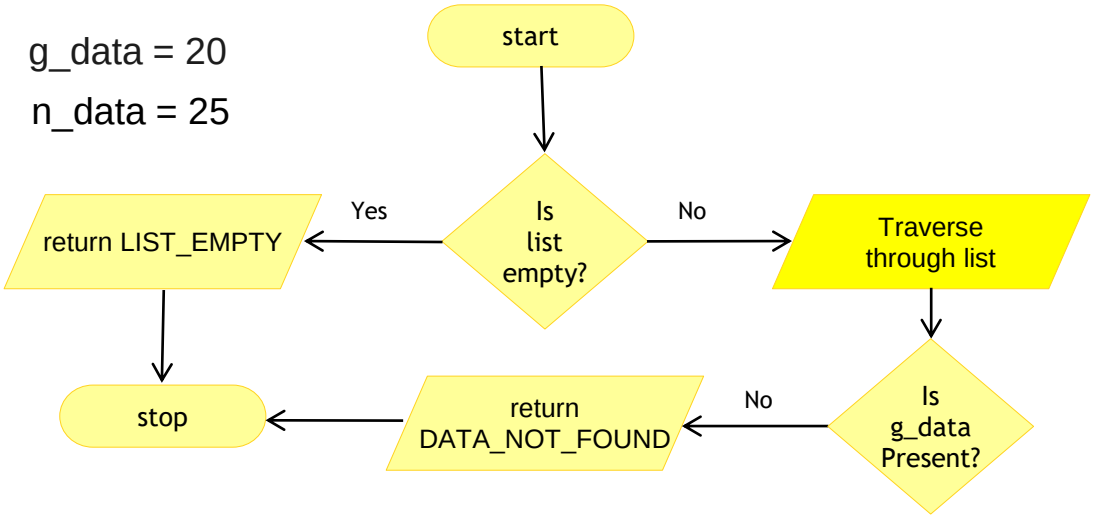


Flowchart
insert_after

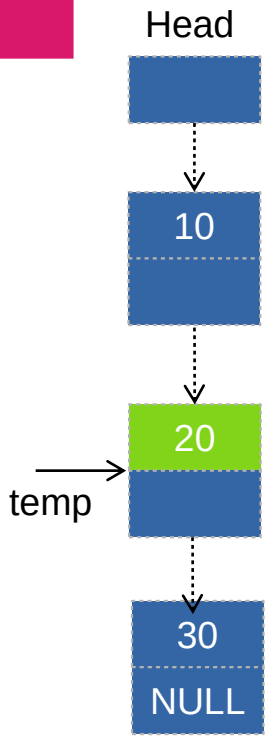


g_data = 20

n_data = 25

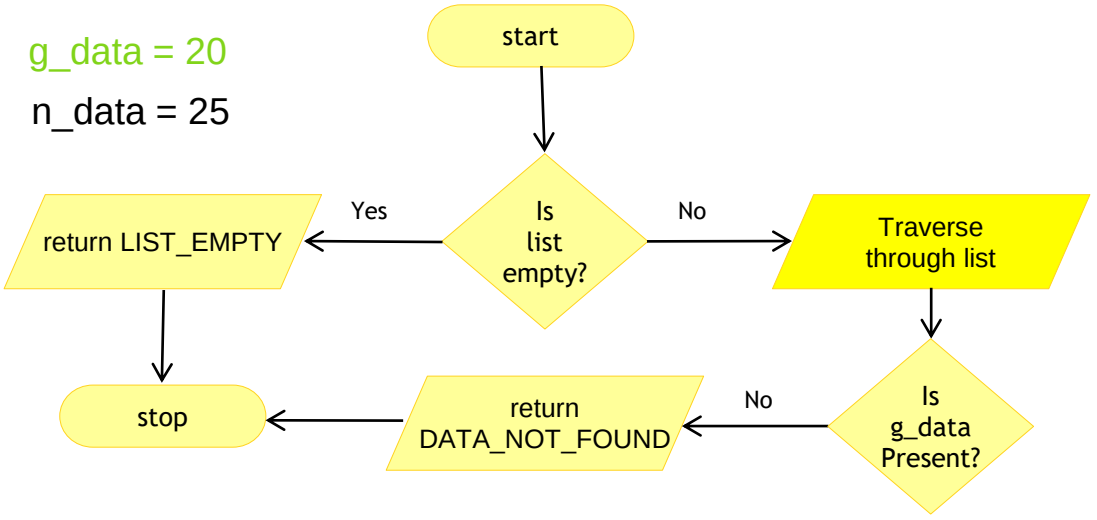


Flowchart
insert_after

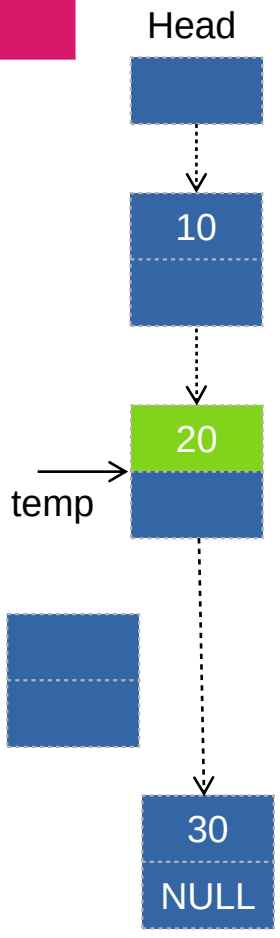


g_data = 20

n_data = 25

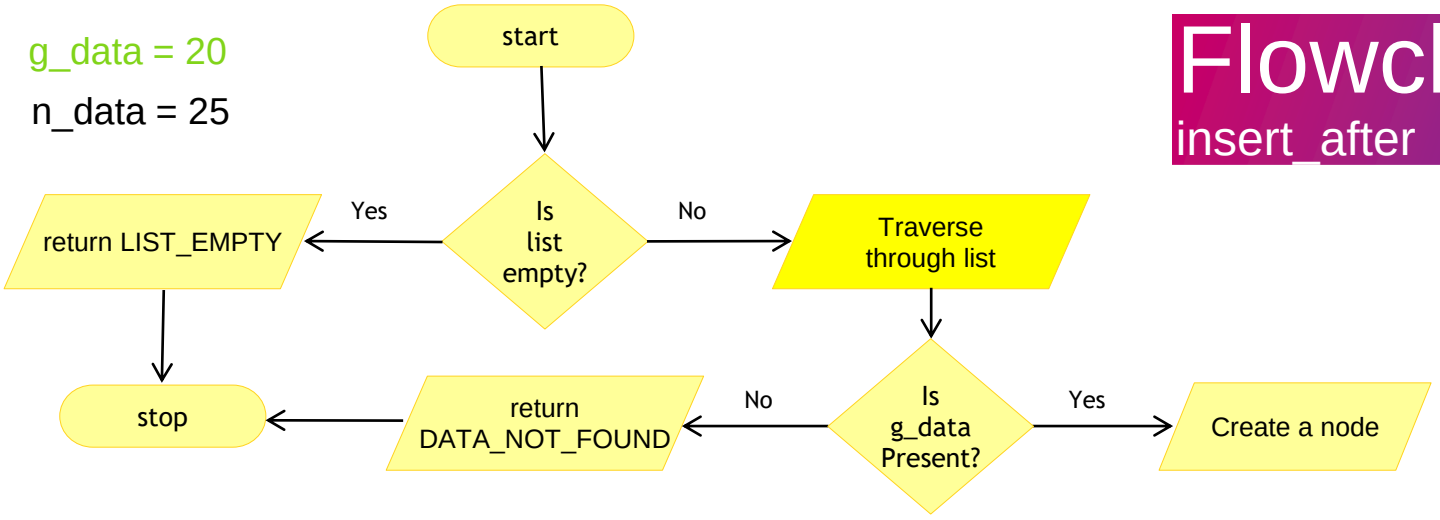


Flowchart
insert_after

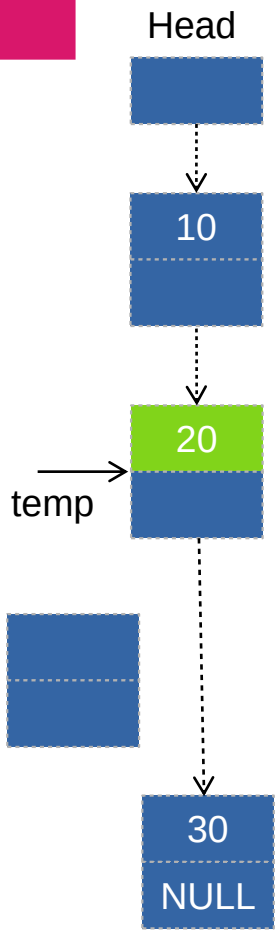


g_data = 20

n_data = 25

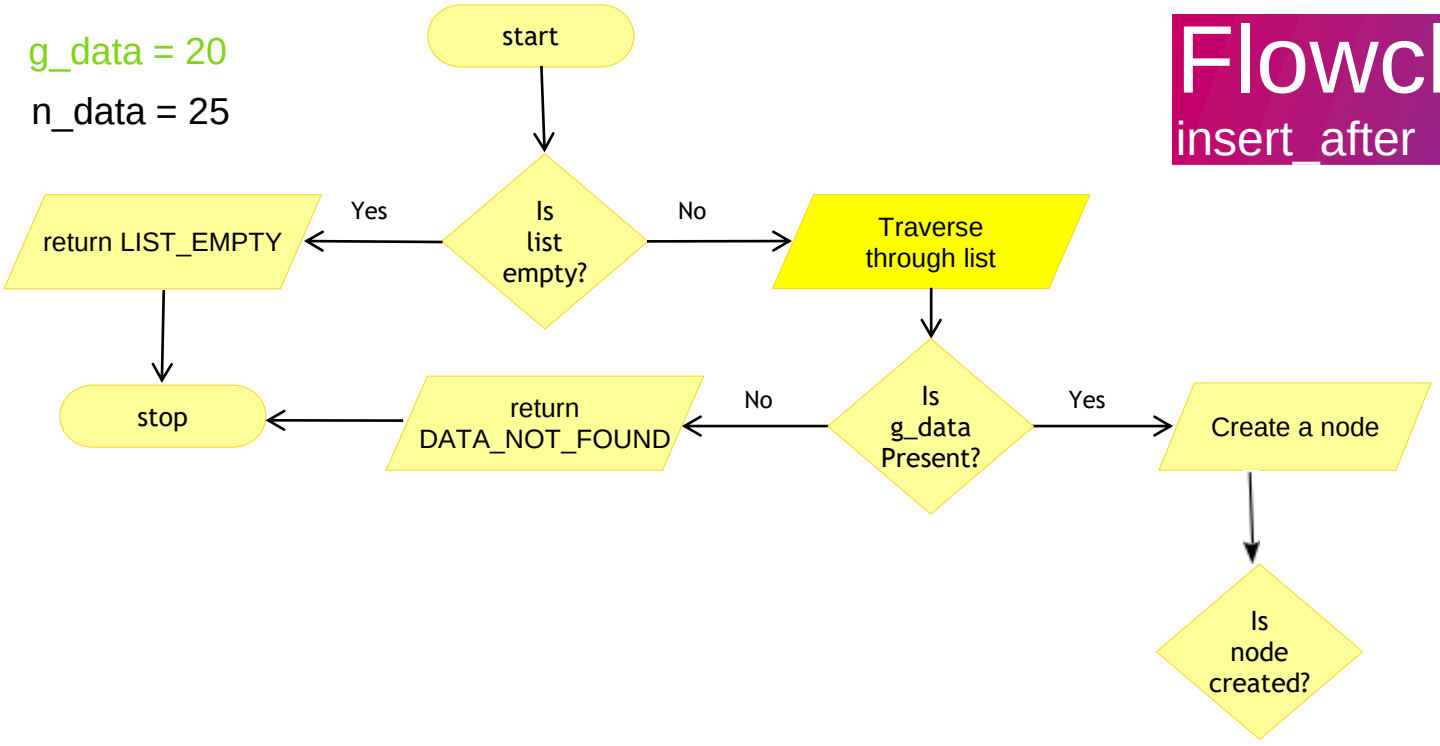


Flowchart
insert_after

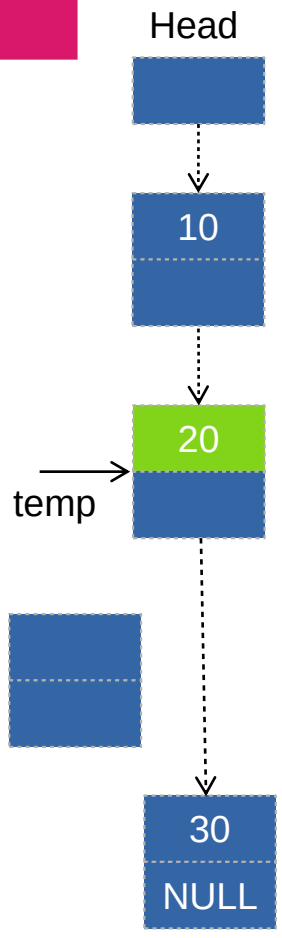


g_data = 20

n_data = 25

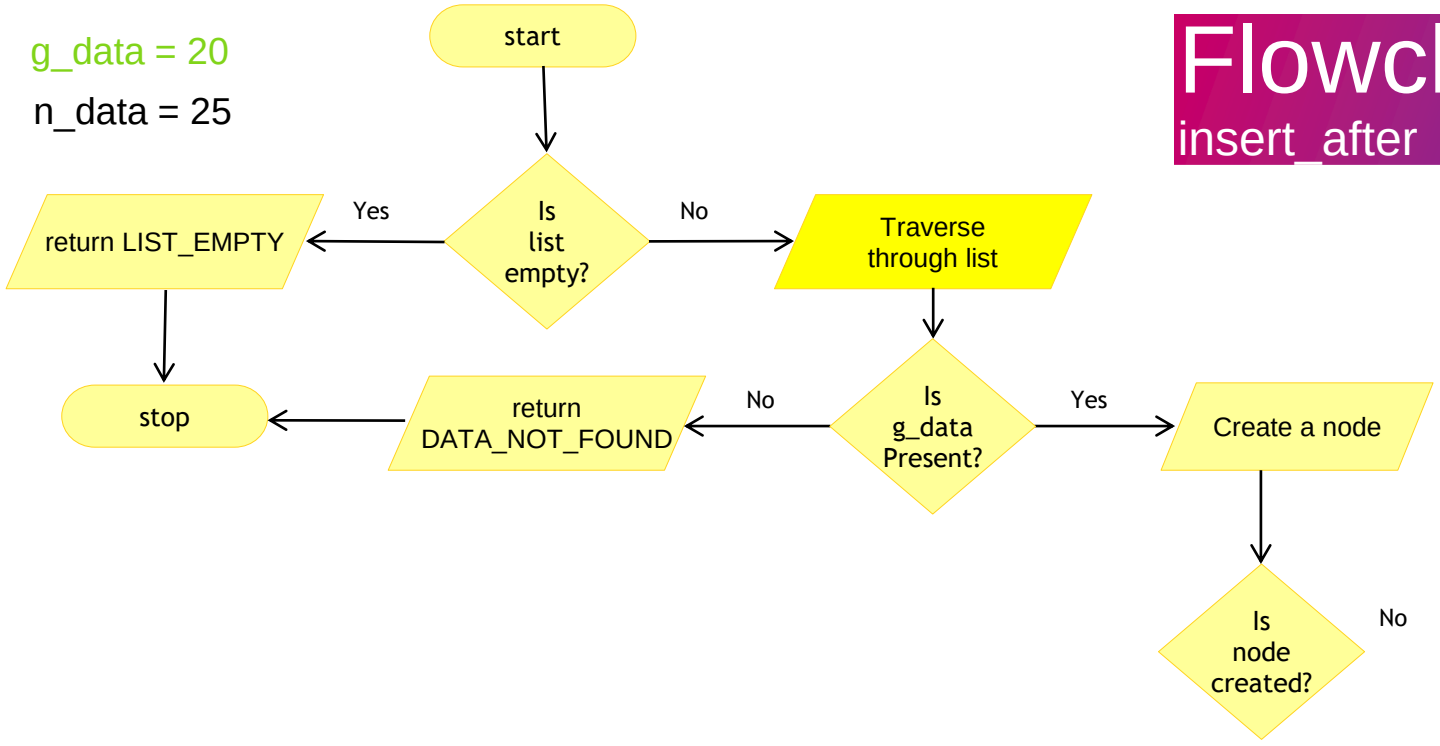


Flowchart
insert_after

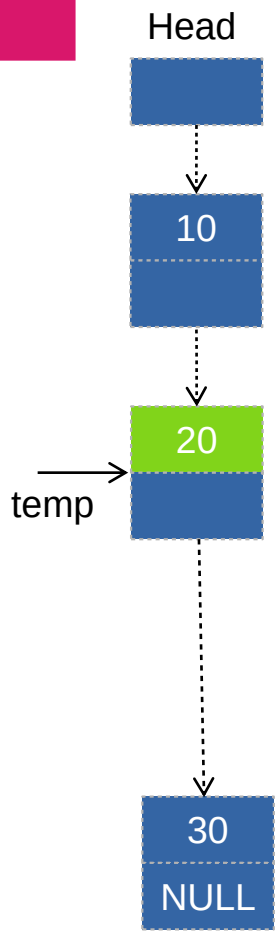


g_data = 20

n_data = 25

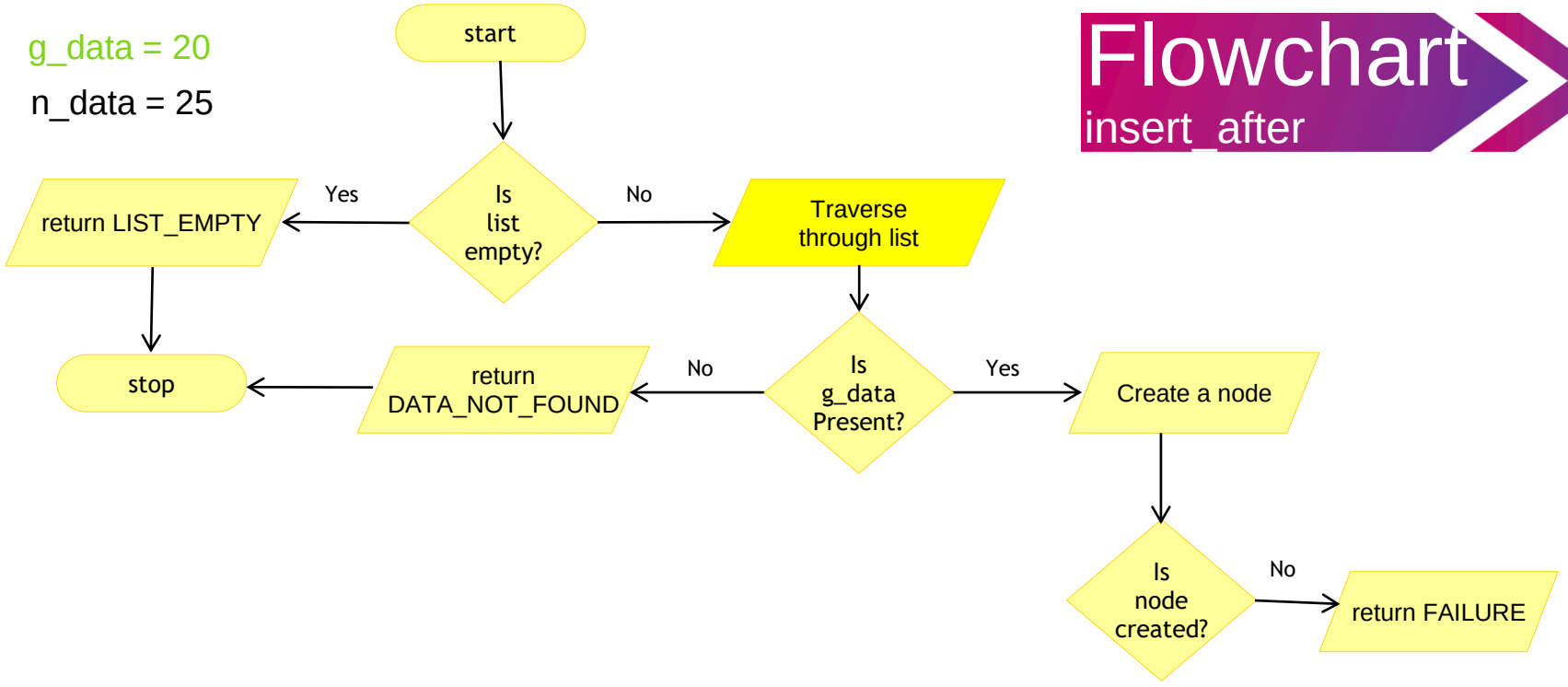


Flowchart
insert_after



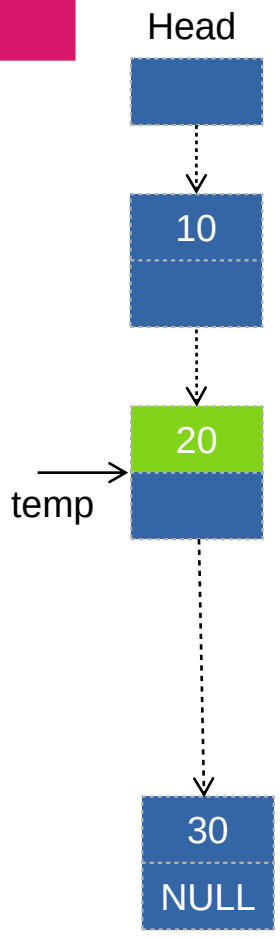
g_data = 20

n_data = 25



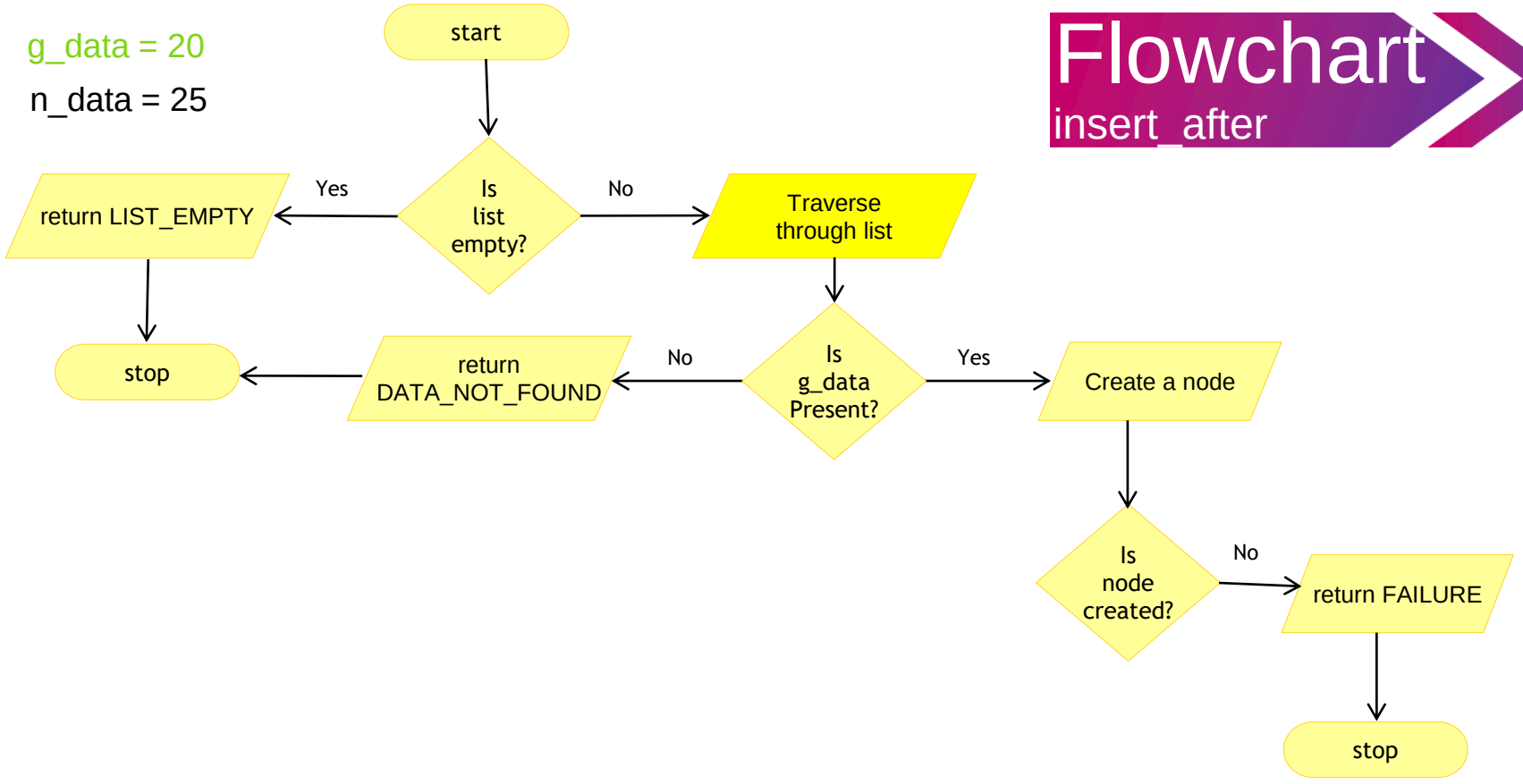
EMERTXE

Flowchart
insert_after



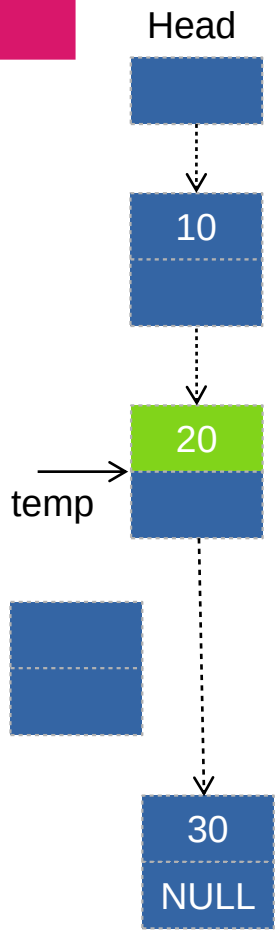
g_data = 20

n_data = 25



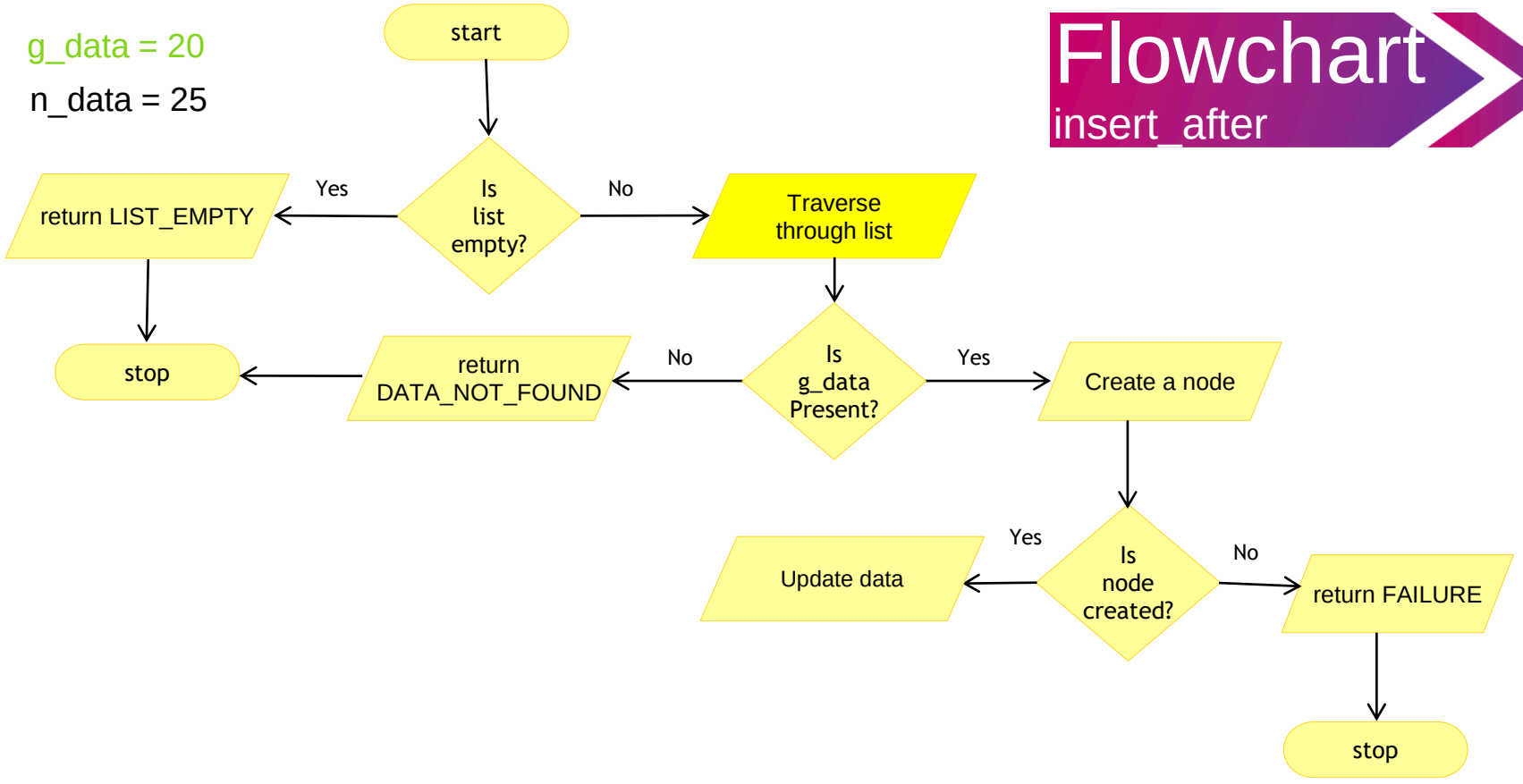
EMERTXE

Flowchart
insert_after

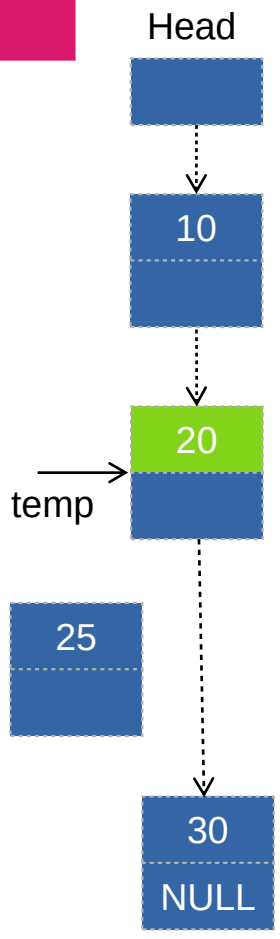


g_data = 20

n_data = 25

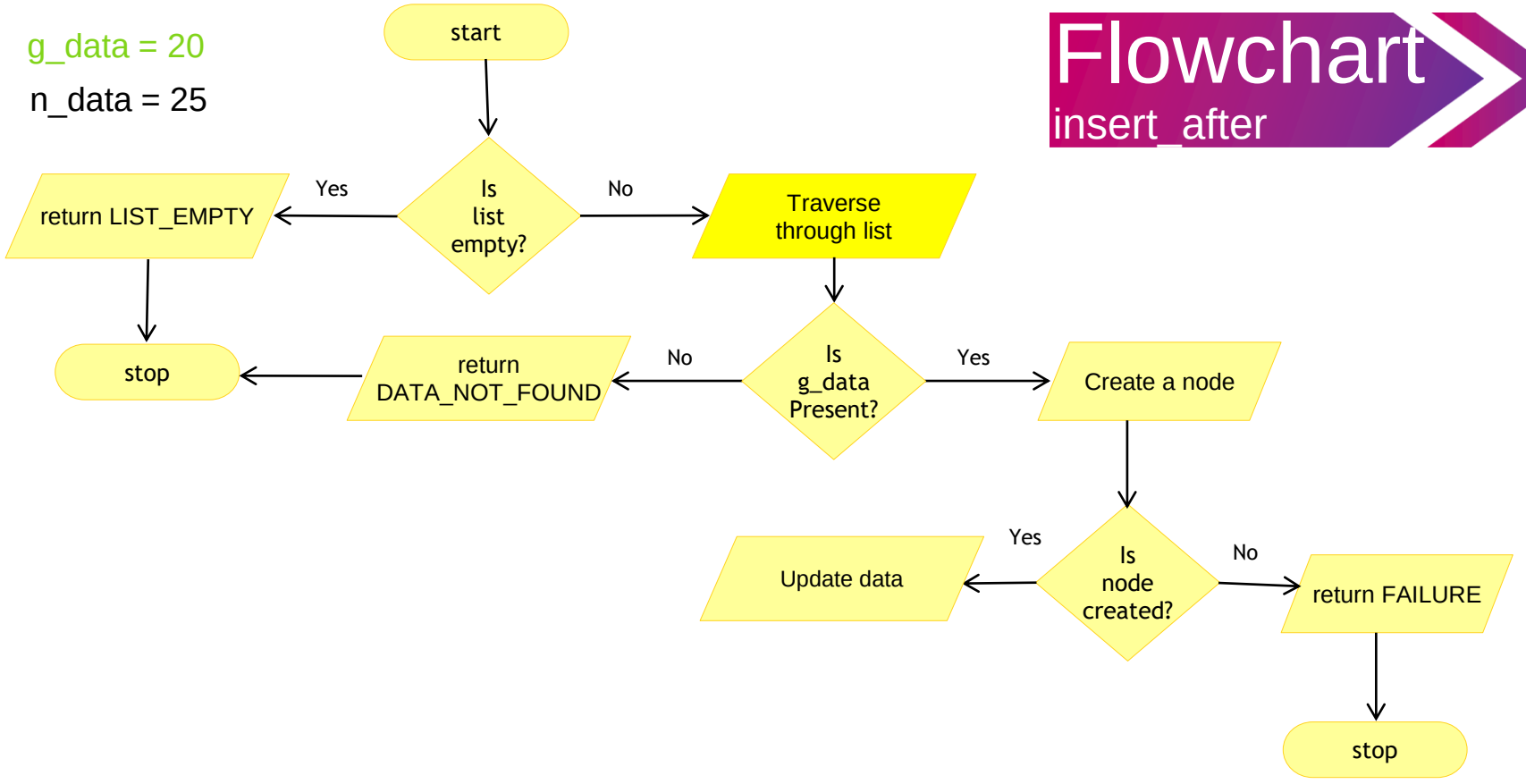


Flowchart
insert_after



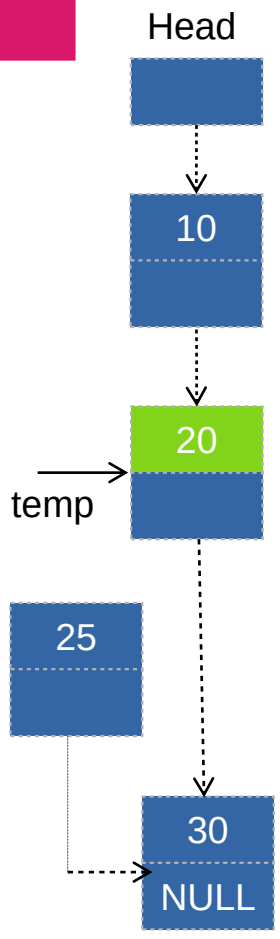
g_data = 20

n_data = 25



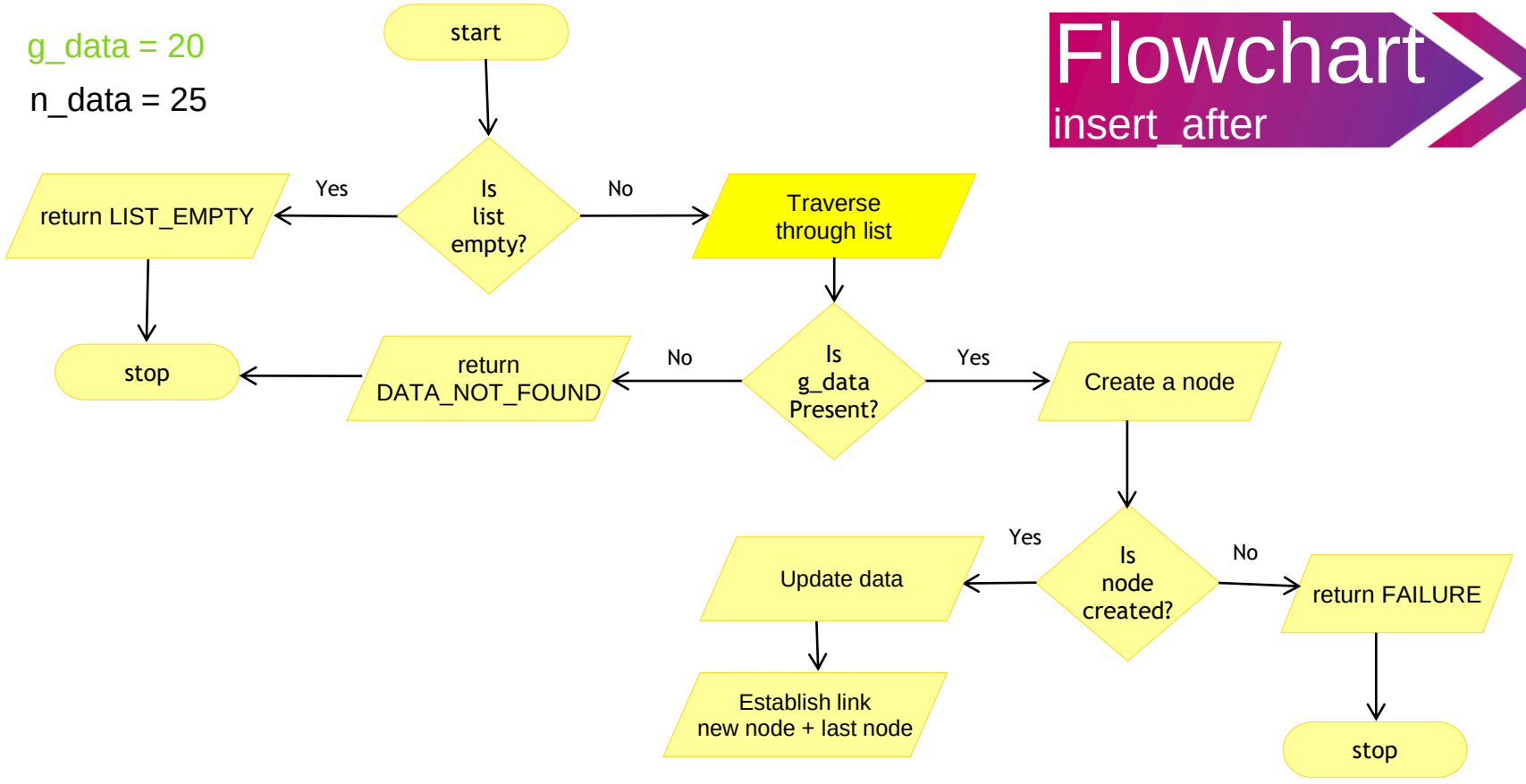
EMERTXE

Flowchart
insert_after



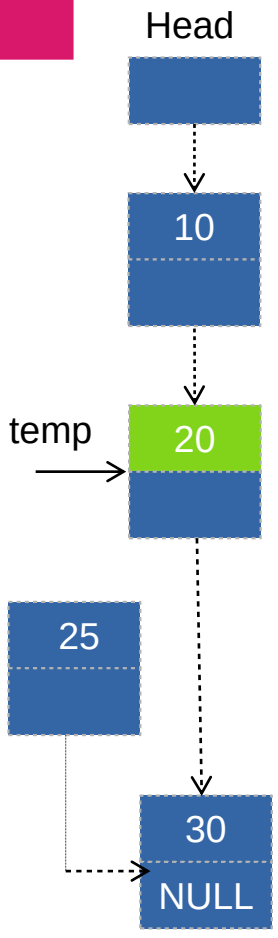
g_data = 20

n_data = 25



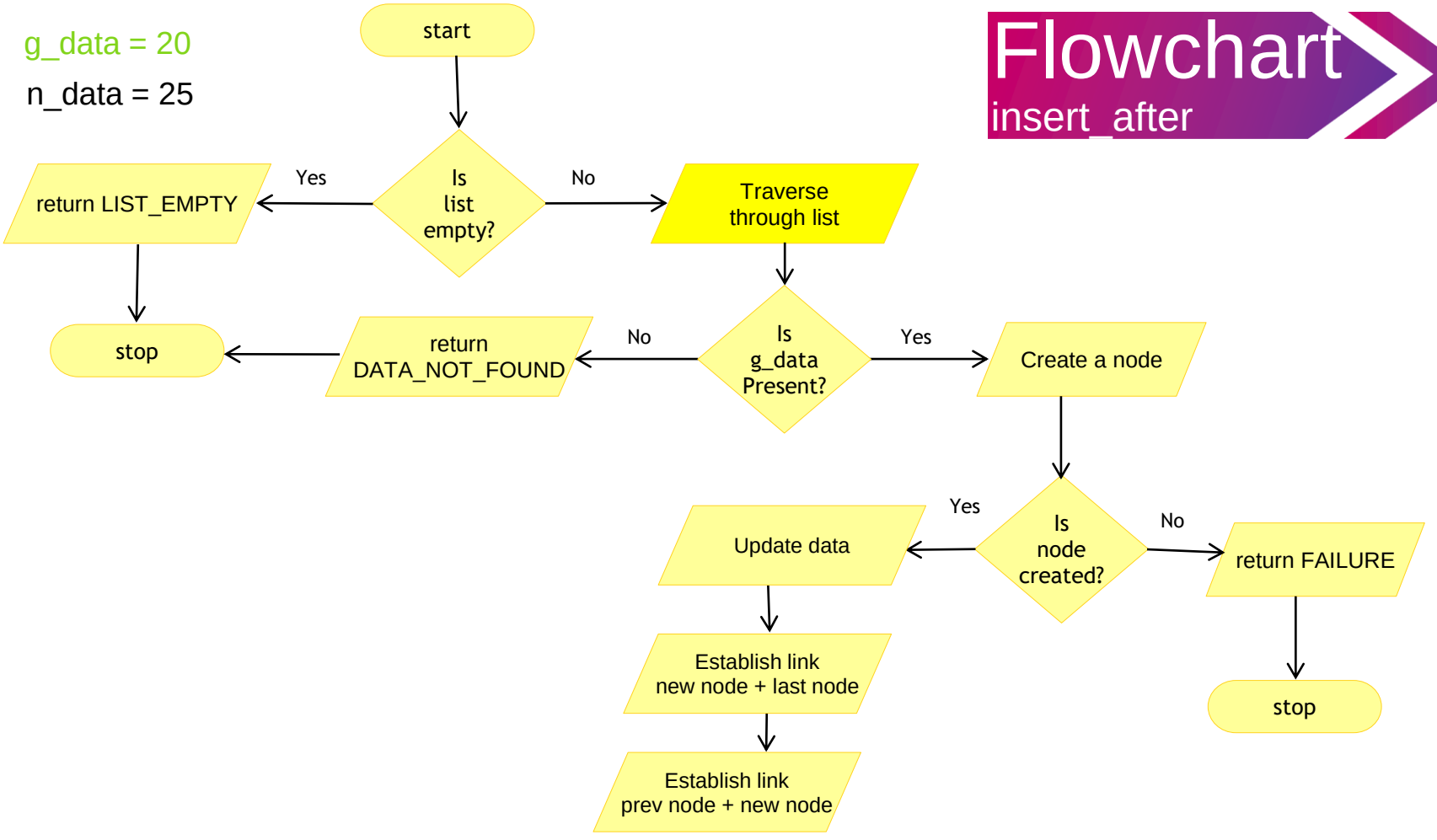
EMERTXE

Flowchart
insert_after



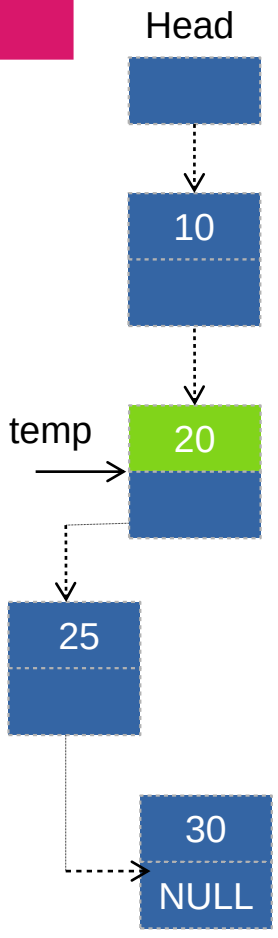
g_data = 20

n_data = 25



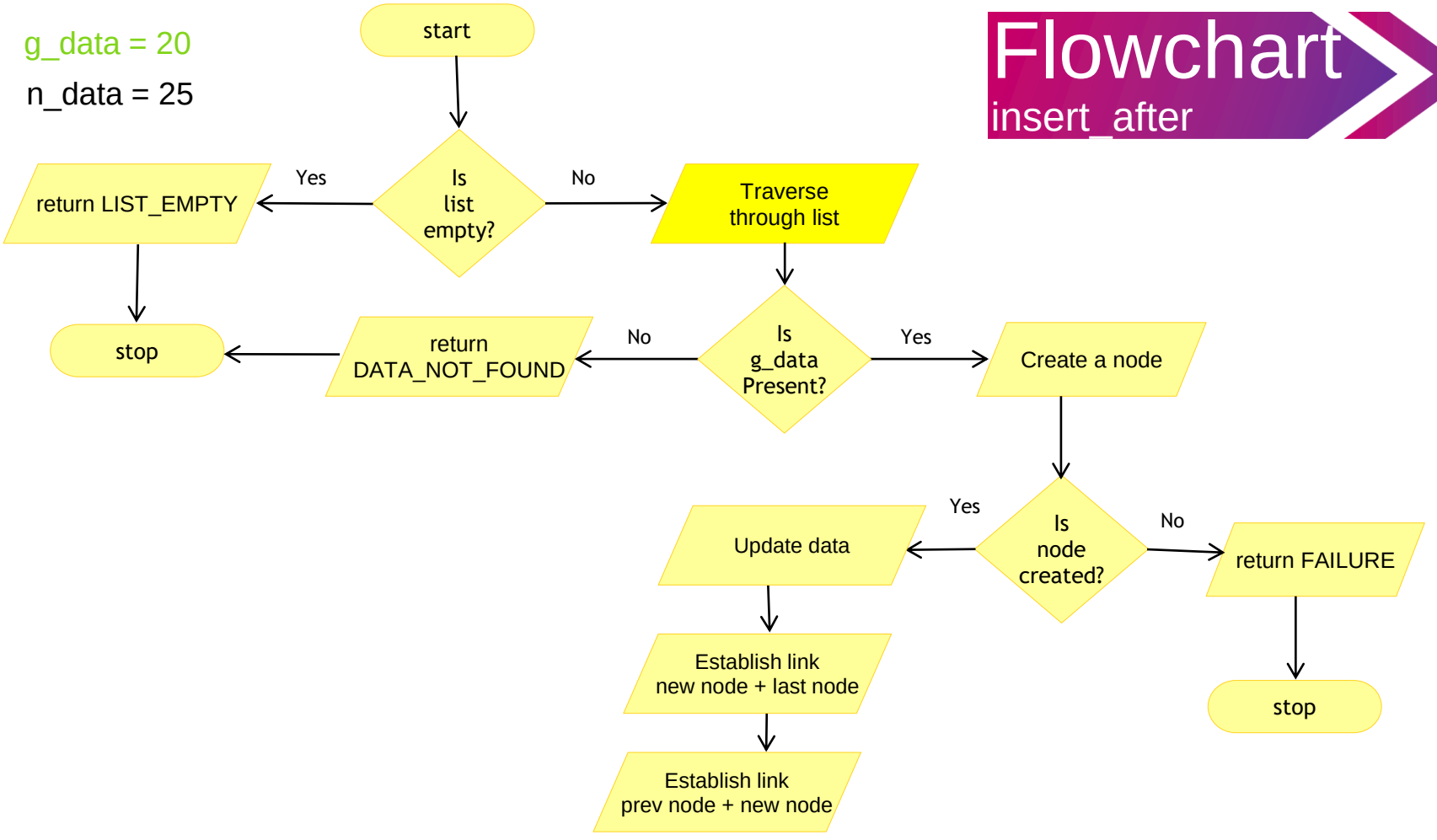
EMERTXE

Flowchart
insert_after



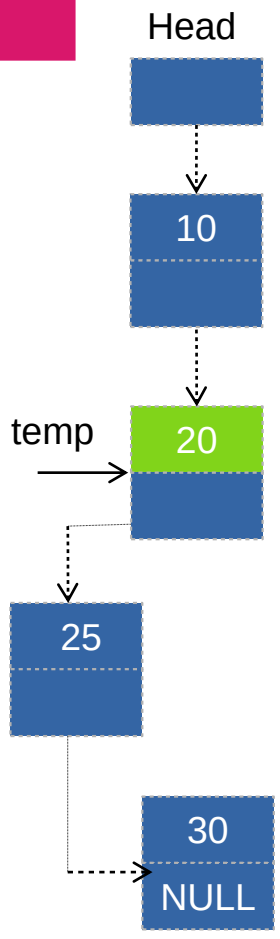
g_data = 20

n_data = 25



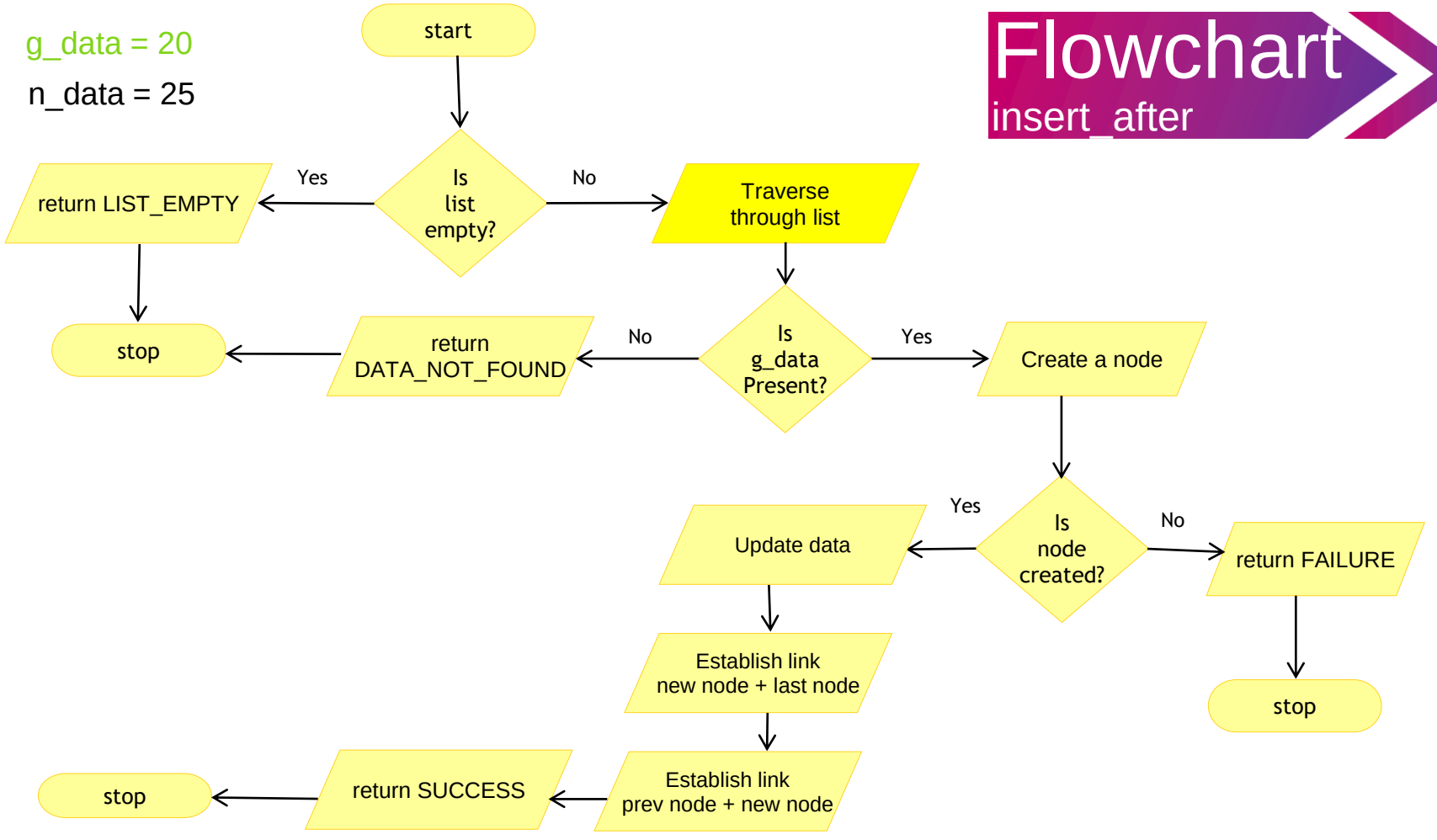
EMERTXE

Flowchart
insert_after



g_data = 20

n_data = 25



EMERTXE

Linked List – Insert after - Algorithm

Algorithm - insert_after

Algorithm
insert_after

Algorithm : Insert_after(Head,g_data,n_data)

1.Input Specification :-

Algorithm
insert_after

Algorithm : Insert_after(Head,g_data,n_data)

1.Input Specification :-

Head : Pointer containing first node address

Algorithm
insert_after

Algorithm : Insert_after(Head,g_data,n_data)

1.Input Specification :-

Head : Pointer containing first node address

g_data : Item after which we wish to insert the n_data

n_data : Item to be added

Algorithm
insert_after

Algorithm : Insert_after(Head,g_data,n_data)

1.Input Specification :-

Head : Pointer containing first node address

g_data : Item after which we wish to insert the n_data

n_data : Item to be added

2.Output Specification :-

Algorithm
insert_after

Algorithm : Insert_after(Head,g_data,n_data)

1.Input Specification :-

Head : Pointer containing first node address

g_data : Item after which we wish to insert the n_data

n_data : Item to be added

2.Output Specification :-

Status : SUCCESS / FAILURE

LIST_EMPTY / DATA_NOT_FOUND

Insert_after(Head,g_data,n_data)

Algorithm
insert_after

Insert_after(Head,g_data,n_data)

Algorithm
insert_after

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)

g_data = 40

n_data = 25

Head

NULL

Algorithm
insert_after

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY

g_data = 40

n_data = 25

Head

NULL

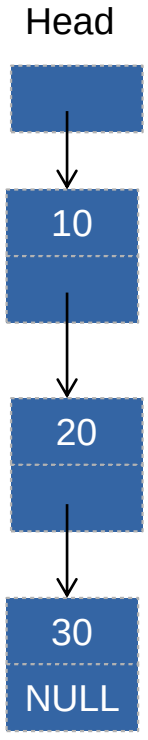
Algorithm
insert_after

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY

g_data = 40

n_data = 25

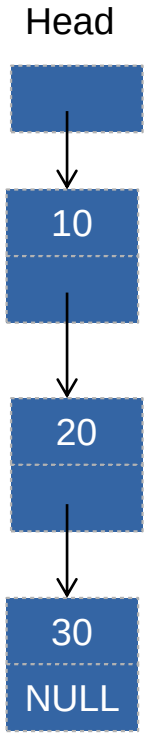


Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY

Algorithm
insert_after

Algorithm
insert_after



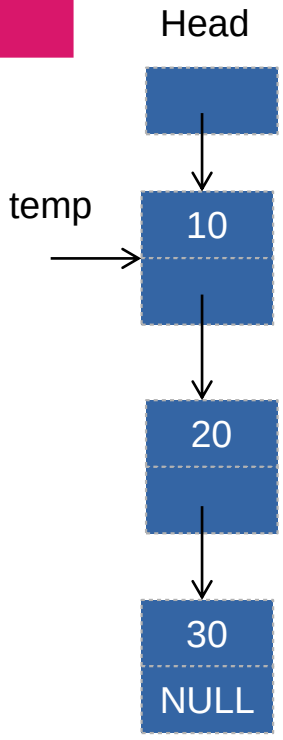
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY

g_data = 40

n_data = 25

EMERTXE



Insert_after(Head,g_data,n_data)

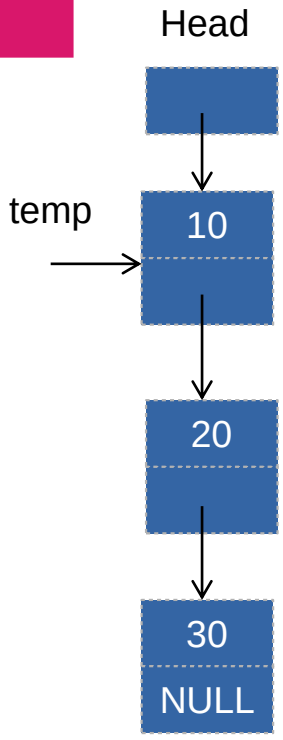
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head

Algorithm
insert_after

g_data = 40

n_data = 25

EMERTXE



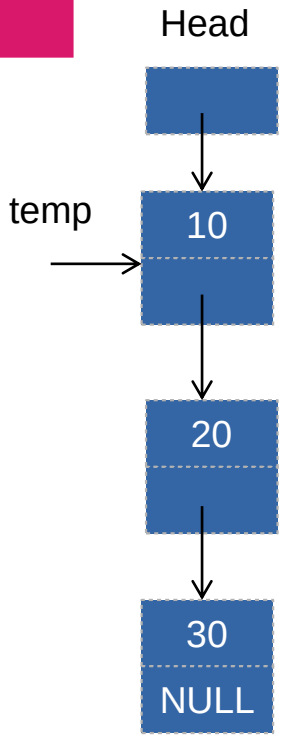
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)

Algorithm
insert_after

g_data = 40

n_data = 25



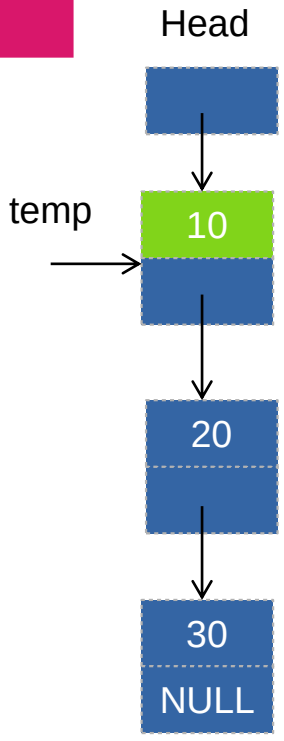
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)

Algorithm
insert_after

g_data = 40

n_data = 25



Algorithm
insert_after

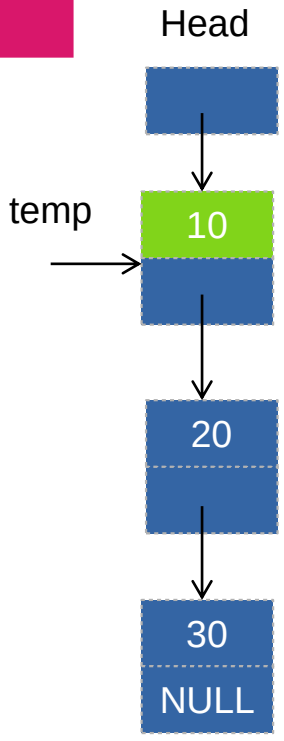
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)

g_data = 40

n_data = 25

EMERTXE



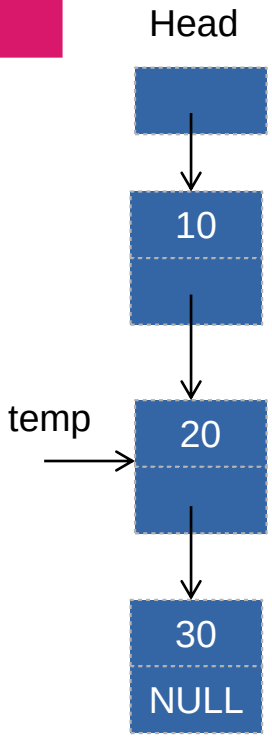
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)

Algorithm
insert_after

g_data = 40

n_data = 25



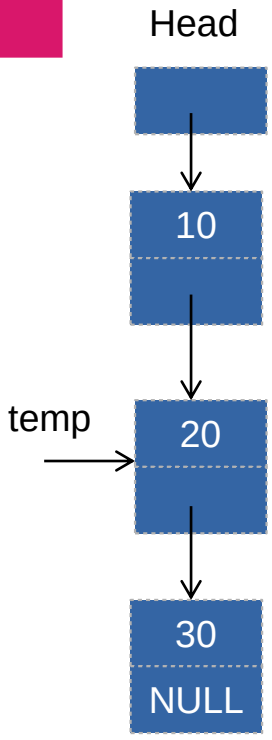
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link

Algorithm
insert_after

g_data = 40

n_data = 25



Algorithm
insert_after

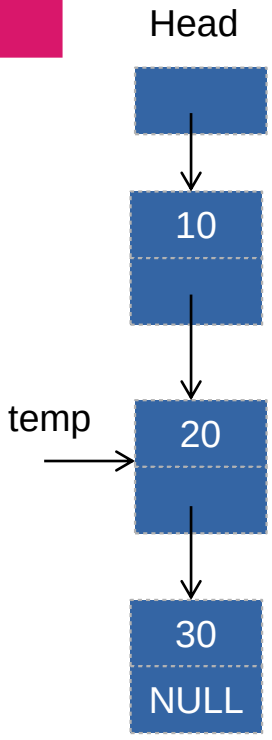
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link

g_data = 40

n_data = 25

EMERTXE



Algorithm
insert_after

Insert_after(Head,g_data,n_data)

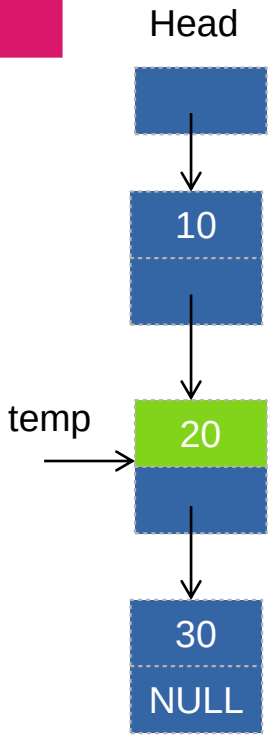
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after

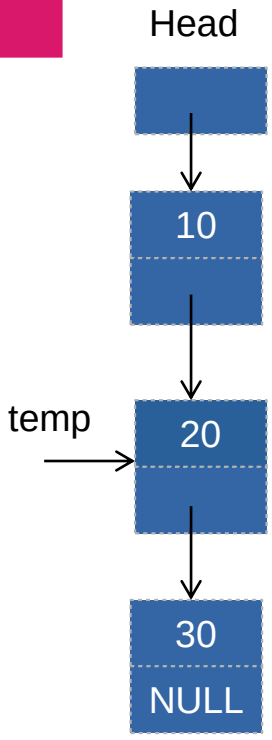


Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → ~~data~~ != g_data)
   temp = temp → link

g_data = 40

n_data = 25



Algorithm
insert_after

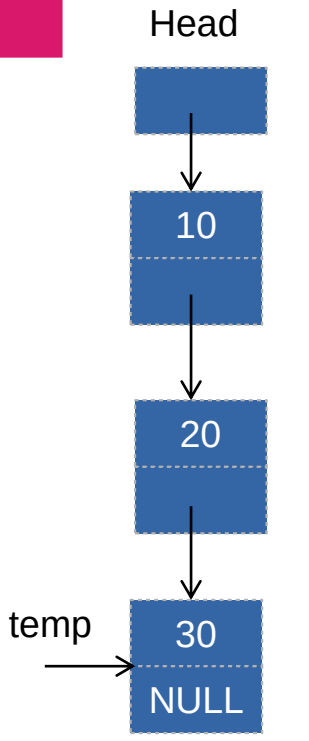
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link

g_data = 40

n_data = 25

EMERTXE



Insert_after(Head,g_data,n_data)

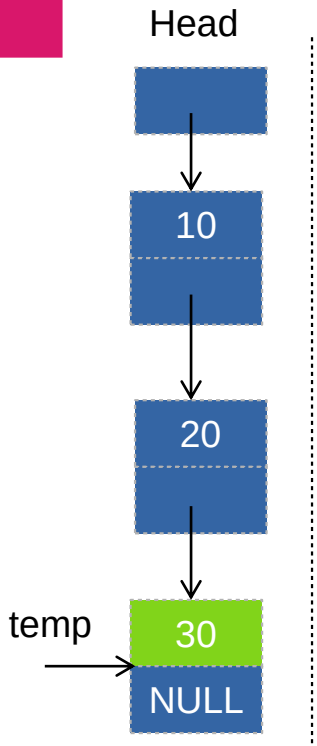
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)
   temp = temp⟶ link

Algorithm
insert_after

g_data = 40

n_data = 25

Algorithm
insert_after



Insert_after(Head,g_data,n_data)

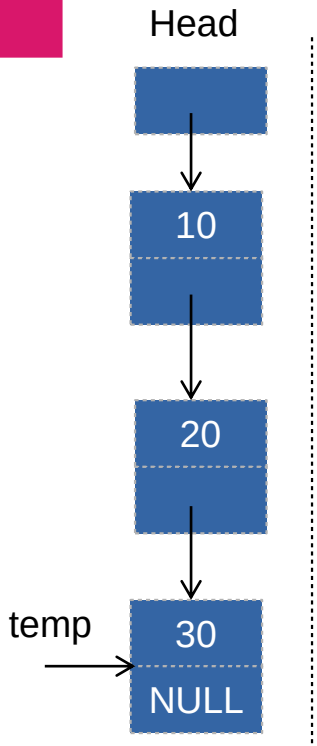
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after



Insert_after(Head,g_data,n_data)

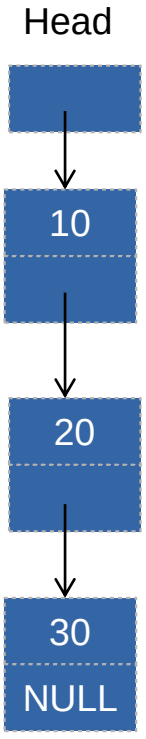
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after



temp = NULL

Insert_after(Head,g_data,n_data)

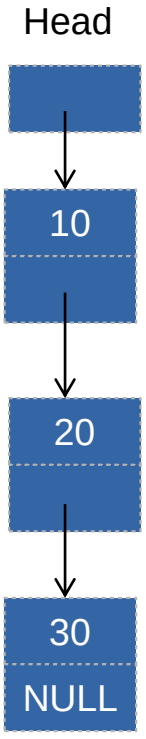
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)
   temp = temp⟶ link

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after



temp = NULL

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)
   temp = temp⟶ link

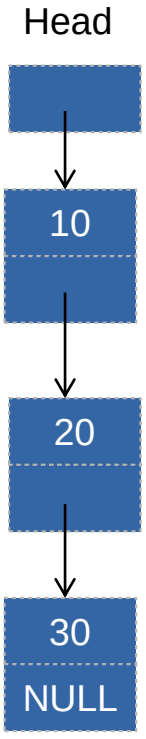
4. return DATA_NOT_FOUND

g_data = 40

n_data = 25

EMERTXE

Algorithm
insert_after



temp = NULL

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶data != g_data)
   temp = temp⟶ link

4. return DATA_NOT_FOUND

g_data = 40

n_data = 25

EMERTXE

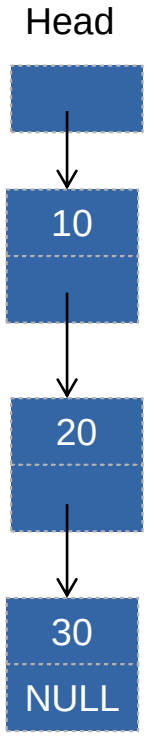
Algorithm
insert_after

Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp⟶ link

g_data = 20

n_data = 25



Algorithm
insert_after

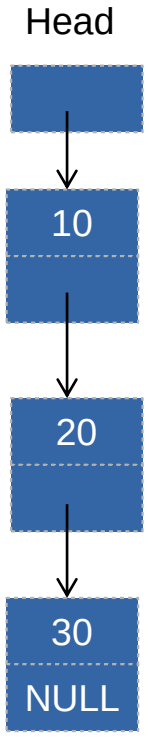
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

Insert_after(Head,g_data,n_data)

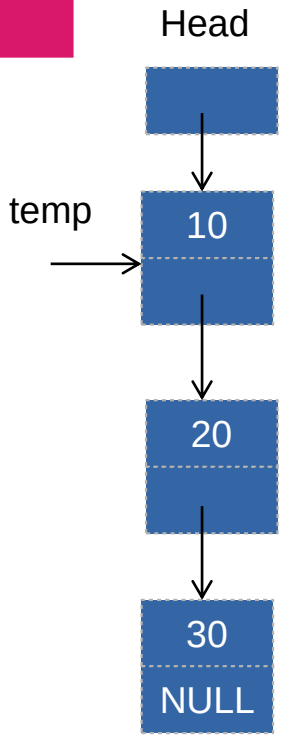
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25

EMERTXE

Algorithm
insert_after



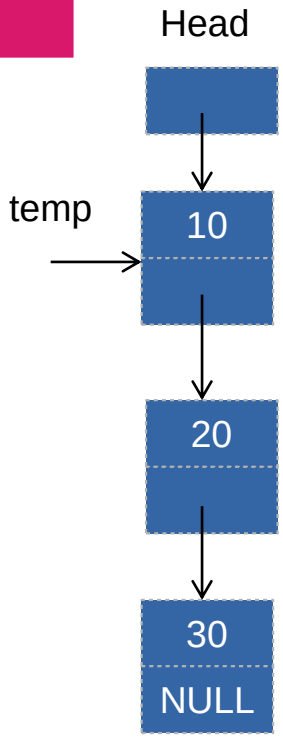
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25

EMERTXE

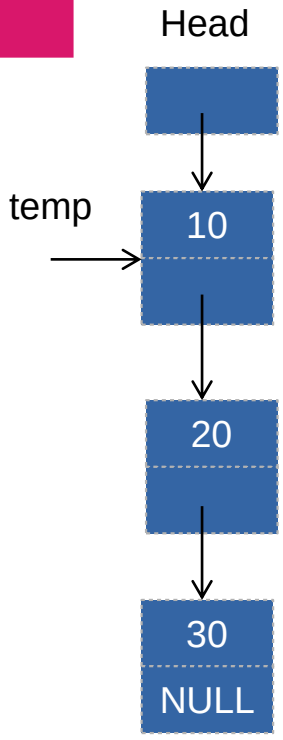


Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25



Insert_after(Head,g_data,n_data)

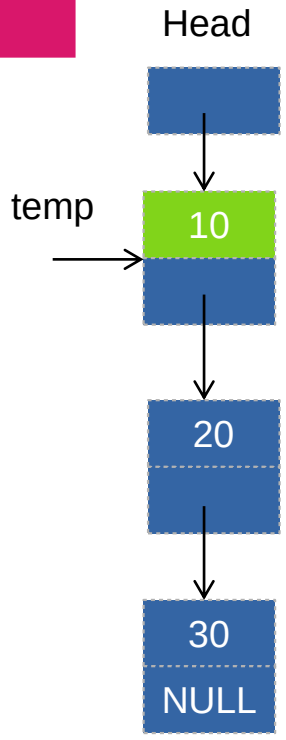
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

Algorithm
insert_after

g_data = 20

n_data = 25

Algorithm
insert_after



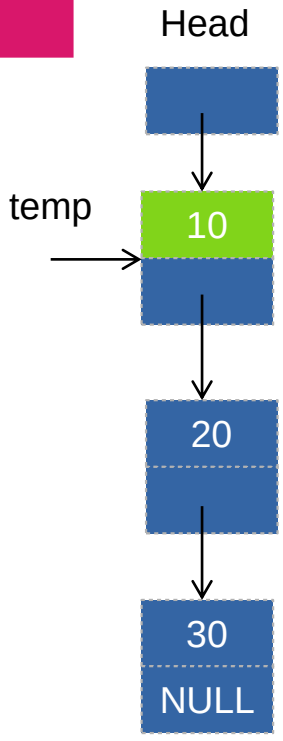
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25

EMERTXE



Insert_after(Head,g_data,n_data)

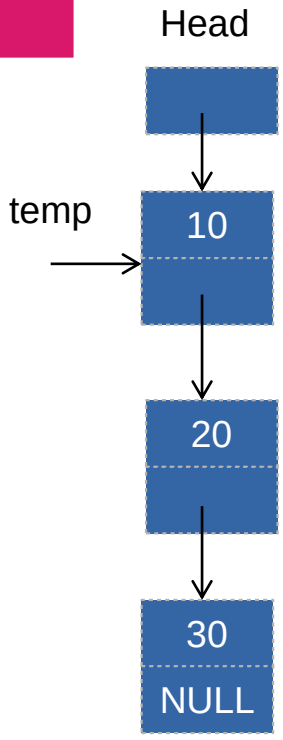
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link

Algorithm
insert_after

g_data = 20

n_data = 25

Algorithm
insert_after



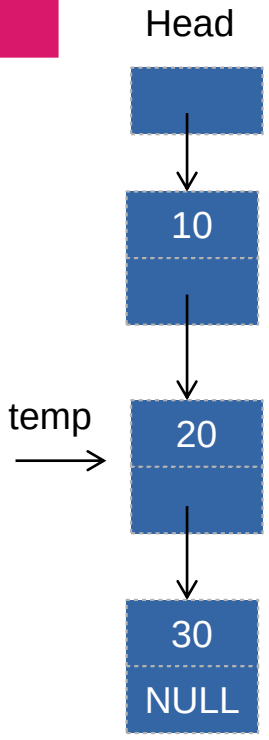
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25

EMERTXE



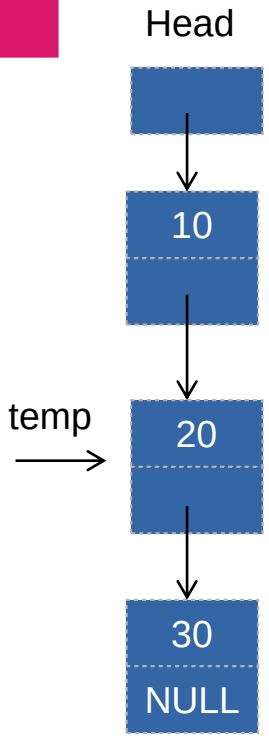
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

Algorithm
insert_after

g_data = 20

n_data = 25



Insert_after(Head,g_data,n_data)

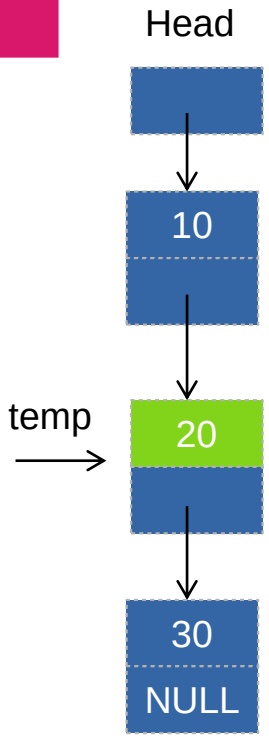
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

Algorithm
insert_after

g_data = 20

n_data = 25

Algorithm
insert_after



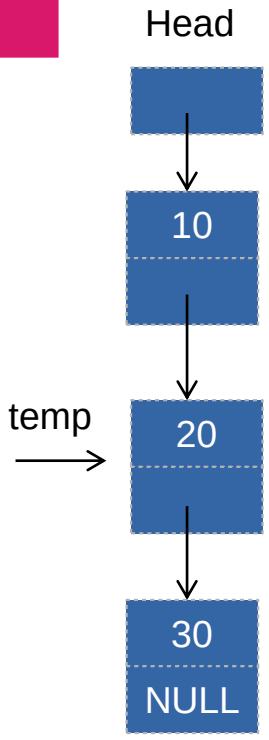
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link

g_data = 20

n_data = 25

Algorithm
insert_after



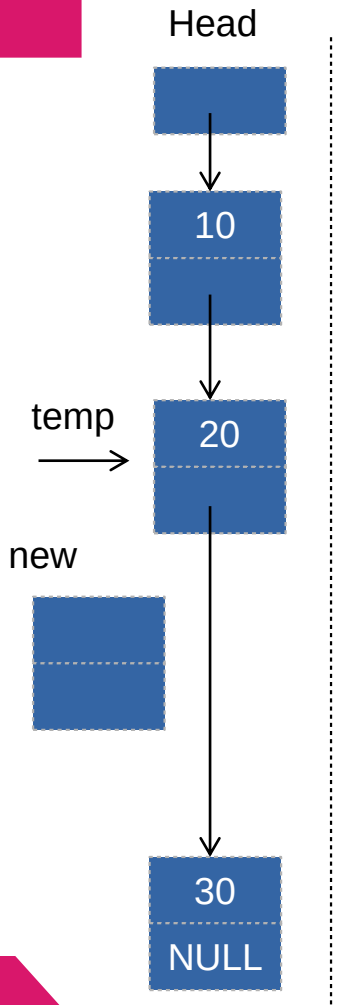
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

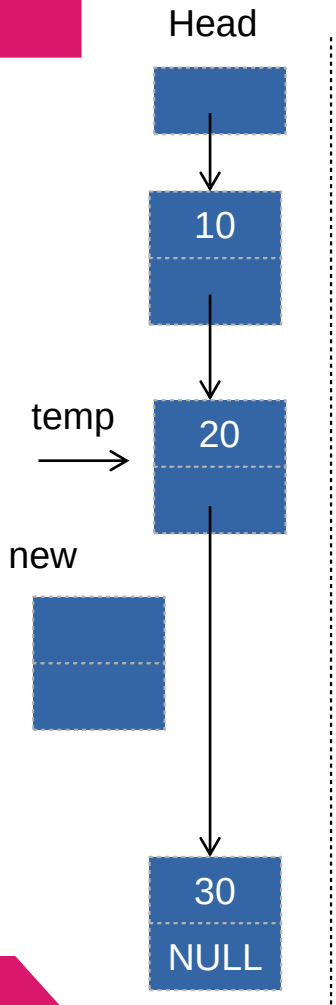
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link
   3.2 else
   new = Memalloc(sizeof (Slist))

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

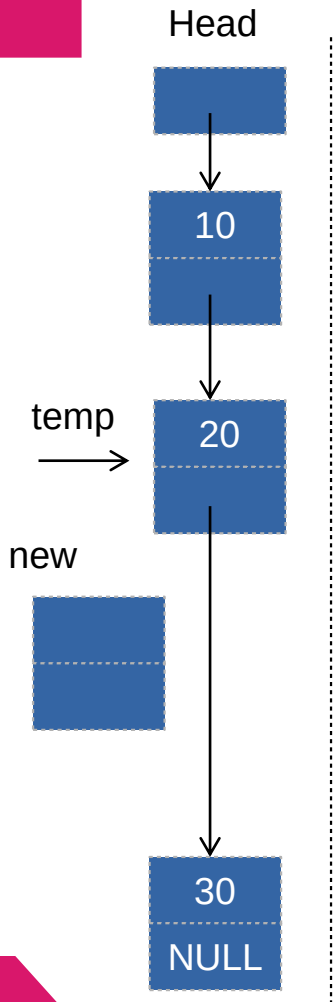
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

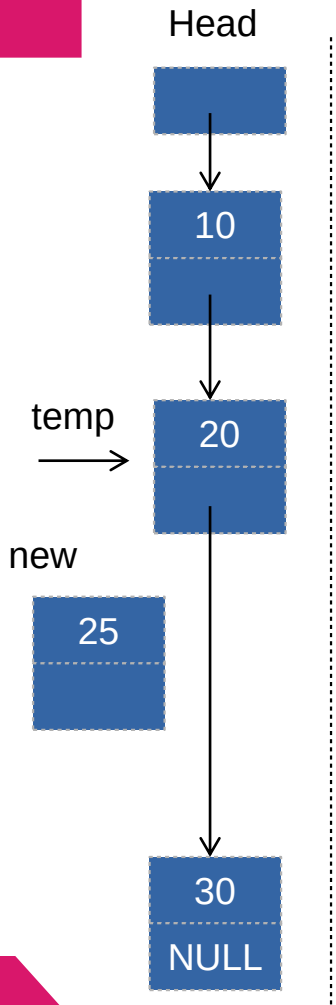
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

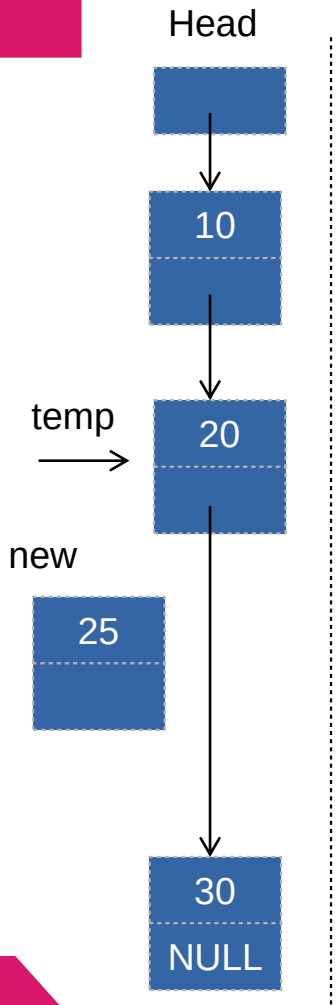
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new → data = n_data

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

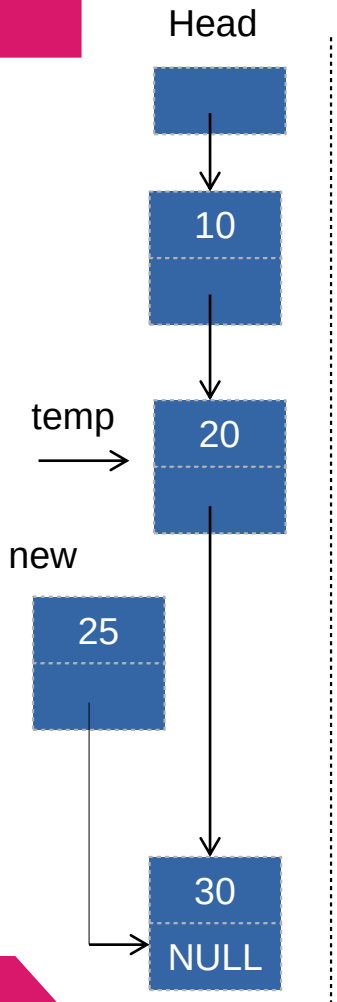
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new → data = n_data
   new → link = temp → link

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

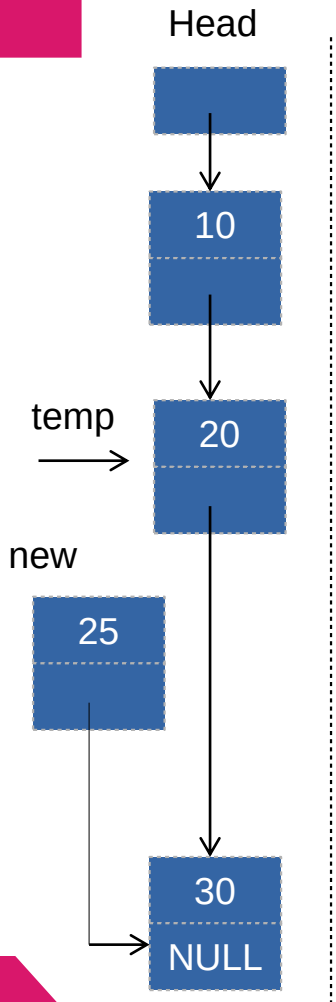
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new → data = n_data
   new → link = temp → link

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

Insert_after(Head,g_data,n_data)

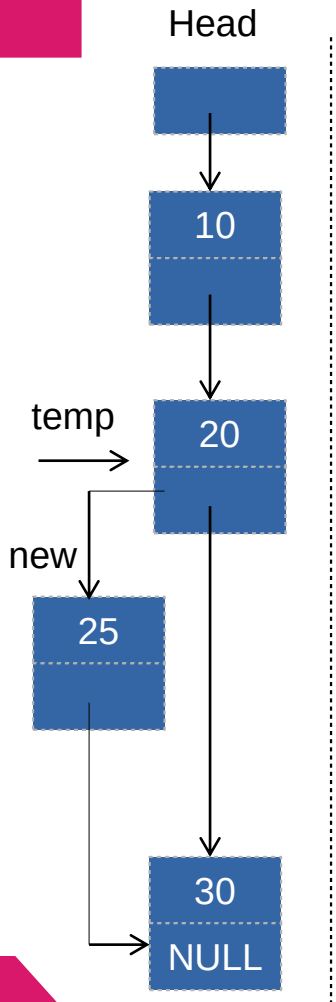
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new → data = n_data
   new → link = temp → link
   temp → link = new

g_data = 20

n_data = 25

EMERTXE

Algorithm
insert_after



Insert_after(Head,g_data,n_data)

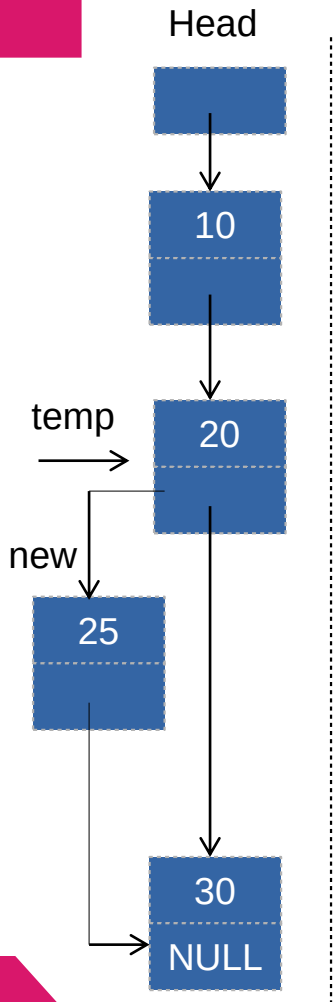
1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new ⟶ data = n_data
   new ⟶ link = temp ⟶ link
   temp ⟶ link = new

g_data = 20

n_data = 25

EMERTXE

Algorithm
insert_after



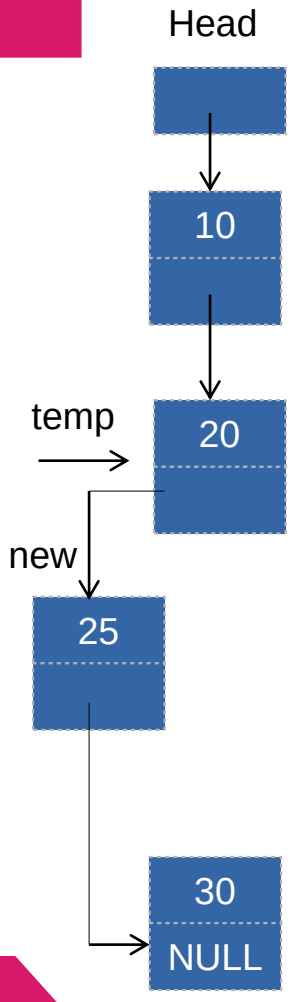
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp ⟶ data != g_data)
   temp = temp ⟶ link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new ⟶ data = n_data
   new ⟶ link = temp ⟶ link
   temp ⟶ link = new

g_data = 20

n_data = 25

EMERTXE



Algorithm
insert_after

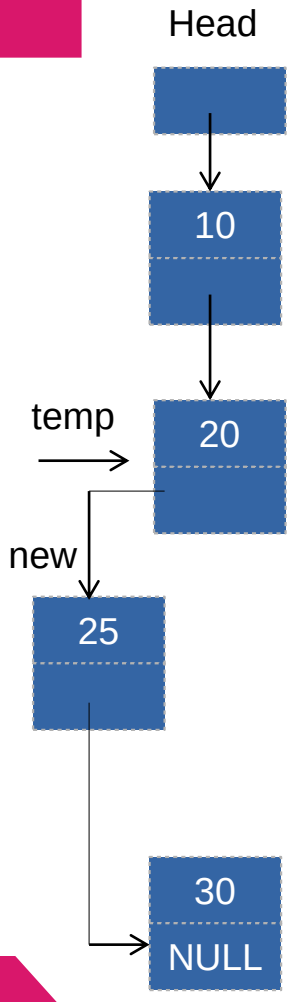
Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new → data = n_data
   new → link = temp → link
   temp → link = new
   return SUCCESS
4. return DATA_NOT_FOUND

g_data = 30

n_data = 35

Algorithm
insert_after



Insert_after(Head,g_data,n_data)

1. if (Head = NULL)
   return LIST_EMPTY
2. temp = Head
3. while (temp != NULL)
   3.1 if (temp → data != g_data)
   temp = temp → link
   3.2 else
   new = Memalloc(sizeof (Slist))
   if (new = NULL)
   return FAILURE
   new → data = n_data
   new → link = temp → link
   temp → link = new
   return SUCCESS
4. return DATA_NOT_FOUND

g_data = 20

n_data = 25

EMERTXE

Code – Insert after(Head,g_data,n_data)