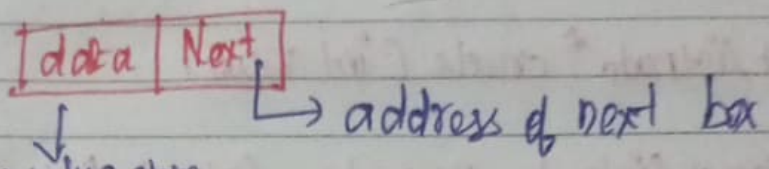
DEC 2026

Single linked list

Single linked list is a chain of boxes

Each box (node) has two parts



In single list you can move only forward Not backward

Program 1:.

write a program to merging two sorted single linked list

Input list 1 [1]→[3]→[5] list 2 [2]→[4]→[6]

output [1]→[2]→[3]→[4]→[5]

we are using structure here because each node in a linked list must store multiple pieces together.

Value
Address

```
# include<stdio.h>
# include <conio.h>
struct listnode {
    int value;
    struct listnode *next;
};
struct listnode* create (int value)
{
    struct listnode newnode = (struct listnode*) malloc(
                              size of(struct listnode*));
    newnode -> value = value;
    newnode -> next = NULL;
    return newnode
}
void PrintList (struct listnode * head)
    while (head != NULL)
    {
        printf ("%d", head->value);
        head = head -> next;
    }
    printf ("NULL\n")
}
// struct listnode merge list( struct listnode *list1,
                               struct listnode *list 2):

    struct listnode dummy;
    dummy.next = NULL;
    struct listnode *current
          = & dummy
```

Listnode

Value
address

Value
next=NULL

List 1

0x1000

Value = 1
next = 0x2000

0x2000

Value = 3
next = 0x3000

0x3000

Value = 5
next = NULL

List 2

0x4000

Value = 2
next = 0x5000

0x5000

Value = 4
next = 0x6000

0x6000

Value = 6
next = NULL

dummy 0x9000

Value = ?
next = NULL

13 Sunday

```
while (list1 != NULL && list2 != NULL)
{
    if (list1 -> value <= list2 -> va
    {
        current -> next = list1
        list1 = list1 -> next;
    }
    else
    {
        current -> next = list2
        list2 = list2 -> next
    }
    current = current -> next
}
if (list1 != NULL)
{
    current -> next = list1
}
else
{
    current -> next = list2
}
return dummy.next;
```