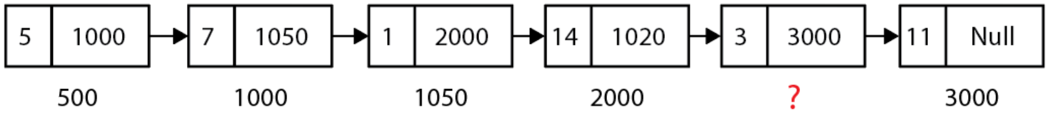
Sum starts at 0.
Temp starts at Head, which has the value 5.
First iteration: Temp->value = 5, so Sum = 0 + 5 = 5. Temp moves to the next node.
Second iteration: Temp->value = 7, so Sum = 5 + 7 = 12. Temp moves to the next node.
Third iteration: Temp->value = 1, so Sum = 12 + 1 = 13. Temp moves to the next node.
Fourth iteration: Temp->value = 14, so Sum = 13 + 14 = 27. Temp moves to the next node.
Now, Temp->Next points to 1020 (where the Next pointer leads to the next node). The loop stops as Temp->Next equals 1020.
The final step: Sum -= Temp->value. Since Temp->value = 14, Sum = 27 - 14 = 13.

Module-5



a. Why do you think linked-list requires more memory than an array when storing the same number of elements?

b. Write down Three Limitations of the array which can be solved by the use of Linked List

c. What is the value of Head? ---> 5

d. What is the value of ? marked address location? --->1020

e. What will be the value of Head->Next->Next->Value? ----->1

f. What will be the value of **Sum** following pseudocode snippets?

```
Sum = 0
Temp = Head
While ( Temp -> Next!= 1020){
      Sum += Temp-> value
      Temp = Temp -> Next
}
Sum -= Temp -> value;
```

sum=27
temp->value=14
temp->next=1020

Sum=27-14=13

a: Linked lists require more memory due to the additional pointers in each node.
b: Limitations of arrays that are solved by linked lists: Fixed size, inefficient insertions/deletions, and memory wastage.
c: The value of Head is 5.
d: The value at the marked address location is 1020.
e: The value of Head->Next->Next->Value is 1.
f: The value of Sum is 27.

a. Why do you think linked-list requires more memory than an array when storing the same number of elements?
Answer: Linked lists require more memory than arrays because each node in a linked list not only stores the value (or data) but also needs extra memory to store a pointer/reference to the next node. For each node, you have the data field and a pointer (next) that requires additional memory.

In an array, each element directly stores the value, and the size of the array is fixed (no need for extra memory for pointers). In contrast, the linked list has extra overhead due to these additional pointers, which increases memory usage.

b. Write down Three Limitations of the array which can be solved by the use of Linked List
Answer:

Fixed Size: Arrays have a fixed size, and resizing an array requires copying data to a new array, which is inefficient. Linked lists can grow or shrink dynamically without the need for resizing.

Inefficient Insertions/Deletions: In arrays, insertions or deletions of elements are costly, especially in the middle of the array, as shifting elements is required. Linked lists allow constant-time insertions and deletions by just updating the pointers.

Memory Wastage: Arrays often allocate memory in advance, leading to unused space if the array isn't fully populated. Linked lists allocate memory for each element individually, so they use memory more efficiently when the exact size is unknown.