Practical no 2 :

```
MAX = 10  # Define maximum size for the hash table

class WOChain:
    def __init__(self):
        # Initialize hash table with -1 (indicating empty slots)
        self.table = [[-1, -1] for _ in range(MAX)]

    def hash_key(self, num):
        return num % MAX

    def chain(self, key, num):
        if all(slot[0] != -1 for slot in self.table):
            print("\nHash Table is Full")
            self.display()
            return

        if self.table[key][0] == -1:
            self.table[key][0] = num
            return

        # Start linear probing to find an empty spot
        current = self.table[key][1]
        if current == -1:
            for i in range((key + 1) % MAX, key):
                if self.table[i][0] == -1:
                    self.table[i][0] = num
                    self.table[key][1] = i
                    return
        else:
            while self.table[current][1] != -1:
                current = self.table[current][1]

            for i in range((current + 1) % MAX, current):
                if self.table[i][0] == -1:
                    self.table[i][0] = num
                    self.table[current][1] = i
                    return

        # If still not placed, try filling from start to key
        for i in range(0, key):
            if self.table[i][0] == -1:
                self.table[i][0] = num
                if current == -1:
                    self.table[key][1] = i
                else:
                    self.table[current][1] = i
                return

        print("No space found to insert.")

    def display(self):
        print("\nThe Hash Table is:")
        print("Index | Value | Link")
        for i in range(MAX):
            print(f"{i:5} | {self.table[i][0]:5} | {self.table[i][1]:4}")

def main():
    h = WOChain()
    try:
        n = int(input("Enter number of elements to insert (<=10): "))
        if n > MAX:
            print("Error: Maximum size of the hash table is 10.")
            return
        for _ in range(n):
            num = int(input("Enter a number to insert: "))
            key = h.hash_key(num)
            h.chain(key, num)
        h.display()
    except ValueError:
        print("Invalid input! Please enter numeric values.")

if __name__ == "__main__":
    main()
```