|  |
| --- |
| **// C Program to delete all occurrences of a given key in linked list**  #include <stdio.h>  #include <stdlib.h>    // A linked list node  struct Node  {      int data;      struct Node \*next;  };    /\* Given a reference (pointer to pointer) to the head of a list     and an int, inserts a new node on the front of the list. \*/  void push(struct Node\*\* head\_ref, int new\_data)  {      struct Node\* new\_node = (struct Node\*) malloc(sizeof(struct Node));      new\_node->data  = new\_data;      new\_node->next = (\*head\_ref);      (\*head\_ref)    = new\_node;  }    /\* Given a reference (pointer to pointer) to the head of a list and     a key, deletes all occurrence of the given key in linked list \*/  void deleteKey(struct Node \*\*head\_ref, int key)  {      // Store head node      struct Node\* temp = \*head\_ref, \*prev;        // If head node itself holds the key or multiple occurrences of key      while (temp != NULL && temp->data == key)      {          \*head\_ref = temp->next;   // Changed head          free(temp);               // free old head          temp = \*head\_ref;         // Change Temp      }        // Delete occurrences other than head      while (temp != NULL)      {          // Search for the key to be deleted, keep track of the          // previous node as we need to change 'prev->next'          while (temp != NULL && temp->data != key)          {              prev = temp;              temp = temp->next;          }            // If key was not present in linked list          if (temp == NULL) return;            // Unlink the node from linked list          prev->next = temp->next;            free(temp);  // Free memory            //Update Temp for next iteration of outer loop          temp = prev->next;      }  }    // This function prints contents of linked list starting from  // the given node  void printList(struct Node \*node)  {      while (node != NULL)      {          printf(" %d ", node->data);          node = node->next;      }  }    /\* Driver program to test above functions\*/  int main()  {      /\* Start with the empty list \*/      struct Node\* head = NULL;        push(&head, 7);      push(&head, 2);      push(&head, 3);      push(&head, 2);      push(&head, 8);      push(&head, 1);      push(&head, 2);      push(&head, 2);        int key = 2; // key to delete        puts("Created Linked List: ");      printList(head);        deleteKey(&head, key);      puts("\nLinked List after Deletion of 1: ");        printList(head);      return 0;  } |