2024-2025 BAHAR DÖNEMİ

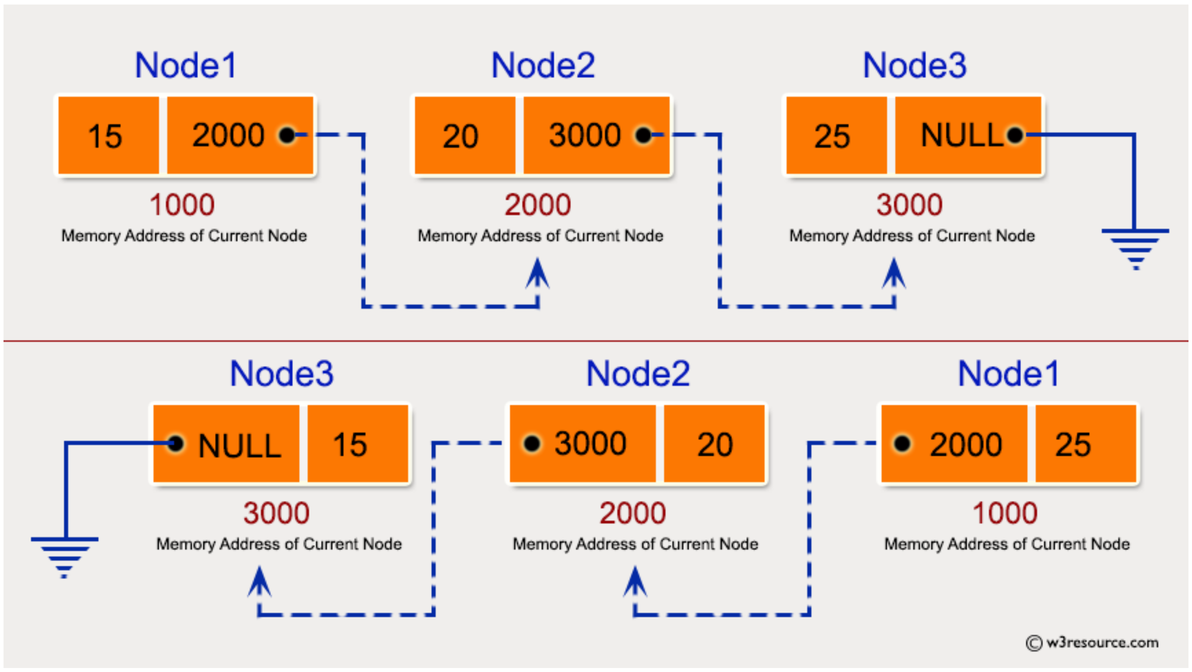
ALGORİTMALAR VE PROGRAMLAMA II

UYGULAMA 5

SORULAR

1. **Soru:**

   n adet düğümden oluşan tek yönlü bağlı liste oluşturan ve bunu ters sırada ekranda görüntüleyen bir C programı yazınız.


Node1
15
2000
1000
Memory Address of Current Node
Node2
20
3000
2000
Memory Address of Current Node
Node3
25
NULL
3000
Memory Address of Current Node
Node3
NULL
15
3000
Memory Address of Current Node
Node2
3000
20
2000
Memory Address of Current Node
Node1
2000
25
1000
Memory Address of Current Node
© w3resource.com

Beklenen ekran çıktısı:

```
Linked List : Create a singly linked list and print it in reverse order :
------------------------------------------------------------------------------
Input the number of nodes : 3
Input data for node 1 : 5
Input data for node 2 : 6
Input data for node 3 : 7

Data entered in the list are :
Data = 5
Data = 6
Data = 7

The list in reverse are :
Data = 7
Data = 6
Data = 5
```

Cevap:

```
#include <stdio.h>
#include <stdlib.h>

// Structure for a node in a linked list
struct node {
    int num;             // Data of the node
    struct node *nextptr;   // Address of the next node
} *stnode;               // Pointer to the starting node

// Function prototypes
void createNodeList(int n);     // Function to create the linked list
void reverseDispList();        // Function to reverse the linked list
void displayList();           // Function to display the linked list

// Main function
int main() {
    int n;

    // Displaying the purpose of the program
    printf("\n\n Linked List : Create a singly linked list and print it in reverse order :\n");
    printf("------------------------------------------------------------------------------\n");

    // Inputting the number of nodes for the linked list
    printf(" Input the number of nodes : ");
    scanf("%d", &n);

    // Creating the linked list with n nodes
    createNodeList(n);
    printf("\n Data entered in the list are : \n");
    // Displaying the data entered in the linked list
    displayList();

    // Reversing the linked list
    reverseDispList();
    printf("\n The list in reverse are :  \n");

    // Displaying the reversed linked list
    displayList();

    return 0;
}

// Function to create a linked list with n nodes
void createNodeList(int n) {
    struct node *fnNode, *tmp;
    int num, i;

    // Allocating memory for the starting node
    stnode = (struct node *)malloc(sizeof(struct node));

    // Checking if memory allocation is successful
    if(stnode == NULL) {
        printf(" Memory can not be allocated.");
    } else {
        // Reading data for the starting node from user input
        printf(" Input data for node 1 : ");
        scanf("%d", &num);
        stnode->num = num;
        stnode->nextptr = NULL; // Setting the next pointer to NULL
        tmp = stnode;
        // Creating n nodes and adding them to the linked list
        for(i = 2; i <= n; i++) {
            fnNode = (struct node *)malloc(sizeof(struct node));

            // Checking if memory allocation is successful
            if(fnNode == NULL) {
                printf(" Memory can not be allocated.");
                break;
            } else {
                // Reading data for each node from user input
                printf(" Input data for node %d : ", i);
                scanf(" %d", &num);

                fnNode->num = num;      // Setting the data for fnNode
                fnNode->nextptr = NULL; // Setting the next pointer to NULL

                tmp->nextptr = fnNode;  // Linking the current node to fnNode
                tmp = tmp->nextptr;     // Moving tmp to the next node
            }
        }
    }
}

// Function to reverse the linked list
void reverseDispList() {
    struct node *prevNode, *curNode;

    if(stnode != NULL) {
        prevNode = stnode;
        curNode = stnode->nextptr;
        stnode = stnode->nextptr;

        prevNode->nextptr = NULL; // Convert the first node as last

        while(stnode != NULL) {
            stnode = stnode->nextptr;
            curNode->nextptr = prevNode;

            prevNode = curNode;
            curNode = stnode;
        }
        stnode = prevNode; // Convert the last node as head
    }
}

// Function to display the linked list
void displayList() {
    struct node *tmp;
    if(stnode == NULL) {
        printf(" No data found in the list.");
    } else {
        tmp = stnode;
        while(tmp != NULL) {
            printf(" Data = %d\n", tmp->num);   // Prints the data of current node
            tmp = tmp->nextptr;                 // Advances the position of current node
        }
    }
}
```