

**Symbiosis Institute of Technology**

**Faculty of Engineering**

**CSE- Academic Year 2024-25**

**Data Structures – Lab Batch 2023-27**

|  |  |
| --- | --- |
| **Lab Assignment No:- 5** | |
|  | |
| **Name of Student** | Pankhuri Varshney |
| **PRN No.** | 23070122160 |
| **Batch** | 2023-27 |
| **Class** | CS-B2 |
| **Academic Year & Semester** | 2024-25  Semester 3 |
| **Date of Performance** | 20th August, 2024 |
|  | |
| **Title of Assignment:** | Menu-driven program for:   1. Creation of One-Way Linked list 2. Insertion at beginning 3. Insertion at end 4. Insertion after specific node 5. Display |
| **Source Code/Algorithm/Flow Chart:** | **Implement a menu-driven program to:**   1. **Create a one-way linked list** 2. **Insert a node at the beginning** 3. **Insert a node at the end** 4. **Insert a node at a specific index** 5. **Display the linked list**   **SOURCE CODE:**  #include <stdio.h>  #include <stdlib.h>  struct node {      int a;      struct node\* ptr;  };  void insertBegin(struct node \*\*head, int val);  void insertEnd(struct node \*\*head, int nextData);  void insertAfterNode(struct node \*\*head, int val, int after);  void printList(struct node\* head);  void insertBegin(struct node \*\*head, int val){      struct node \*newNode=(struct node\*)malloc(sizeof(struct node\*));      newNode->a=val;      newNode->ptr=NULL;      if(\*head==NULL){          \*head=newNode;      }      else{          newNode->ptr=\*head;          \*head=newNode;      }  }  void insertEnd(struct node \*\*head, int nextData){      struct node \*newNode=(struct node\*)malloc(sizeof(struct node));      newNode->a=nextData;      newNode->ptr=NULL;      if(\*head==NULL){          \*head=newNode;      }      else{          struct node \*temp=\*head;          while(temp->ptr!=NULL){              temp=temp->ptr;          }          temp->ptr=newNode;      }  }  void insertAfterNode(struct node \*\*head, int val, int after){      struct node \*newNode=(struct node\*)malloc(sizeof(struct node\*));      newNode->a=val;      if(\*head==NULL){          printf("INVALID! LIST IS EMPTY\n");      }      else{          struct node \*temp=\*head;          while(temp->ptr!=NULL && temp->a!=after){              temp=temp->ptr;          }          if(temp->a==after){              newNode->ptr=temp->ptr;              temp->ptr=newNode;          }          else{              printf("INVALID OPERATION! INDEX OUT OF BOUND\n");          }      }  }  void printList(struct node\* head){      struct node \*temp=head;      if(head==NULL)          printf("LINKED LIST EMPTY\n");      int i=1;      while(temp!=NULL){          printf("%d\t",temp->a);          temp=temp->ptr;      }      printf("\n");  }  int main(){      struct node \*head=NULL;      while(1){          printf("Enter 1: to create linked list\nEnter 2: to insert at beginning\nEnter 3: to insert at the end\nEnter 4: to insert after a specific node\nEnter 5: to display list\nEnter 6: to exit\nEnter your choice: ");          int ch;          scanf("%d", &ch);          switch(ch){              case 1:{                  printf("Enter the initial size of Linked List: ");                  int size;                  scanf("%d", &size);                  for(int i=0; i<size; i++){                      printf("Enter a node value: ");                      int val;                      scanf("%d", &val);                      insertEnd(&head, val);                  }                  break;              }              case 2:{                  printf("Enter a number: ");                  int num;                  scanf("%d", &num);                  insertBegin(&head, num);                  break;              }              case 3:{                  printf("Enter a number: ");                  int num;                  scanf("%d", &num);                  insertEnd(&head, num);                  break;              }              case 4:{                  printf("Enter a number: ");                  int num, index;                  scanf("%d", &num);                  printf("Enter a node after which the number will be added: ");                  scanf("%d", &index);                  insertAfterNode(&head, num, index);                  break;              }              case 5:{                  printList(head);                  break;              }              case 6:{                  goto end;              }              default:              printf("INVALID CHOICE\n");          }      }      end: printf("\n");      return 0;  } |
| **Output Screenshots** | 1. **CREATION OF LINKED LIST AND DISPLAY:**      1. **INSERTION AT BEGINNING AND DISPLAY:**      1. **INSERTION AT END AND DISPLAY:**      1. **INSERTION AT INDEX 2 AND DISPLAY** |
| **Practice questions** | 1. Create a structure called "Student" with members name, age, and total marks. Write a C program to input data for two students, display their information, and find the average of total marks. 2. Define a structure named Time with members hours, minutes, and seconds. Write a C program to input two times, add them, and display the result in proper time format. 3. Create a structure named Book to store book details like title, author, and price. Write a C program to input details for three books, find the most expensive and the lowest priced books, and display their information. 4. Write a program in C to add numbers using call by reference. 5. Write a program in C to find the maximum number between two numbers using a pointer. |
| **Conclusion** | Thus, we have studied the concept of Linked List and how it is different from arrays. |