SANGHAMITRA R
1BM22CS237
LAB 6

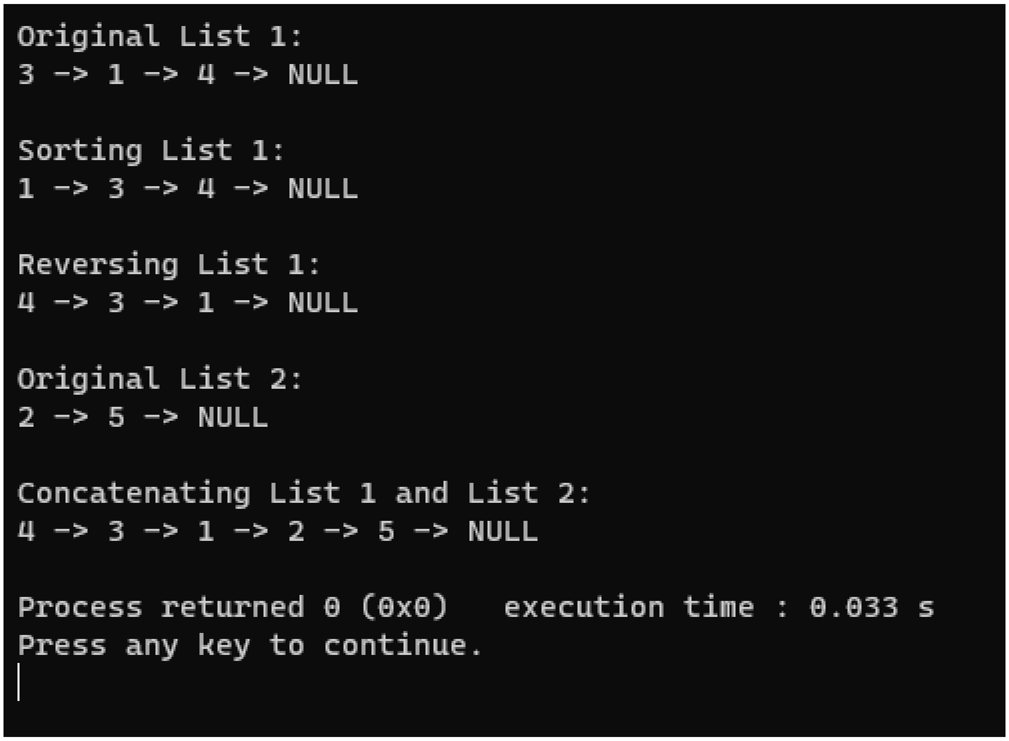
1. 6a) WAP to Implement Single Link List with following operations: Sort the linked list, Reverse the linked
list, Concatenation of two linked lists.

Code:

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

struct Node {
    int data;
    struct Node *next;
};

typedef struct Node Node;

Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

void append(Node** head, int data) {
    Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}

void display(Node* head) {
    Node* current = head;
    while (current != NULL) {
        printf("%d -> ", current->data);
        current = current->next;
    }
    printf("NULL\n");
}

void sortList(Node** head) {
    if (*head == NULL) {
        return;
    }

    int temp;
    Node* current1 = *head;
    Node* current2;

    while (current1 != NULL) {
        current2 = current1->next;

        while (current2 != NULL) {
            if (current1->data > current2->data) {
                temp = current1->data;
                current1->data = current2->data;
                current2->data = temp;
            }

            current2 = current2->next;
        }

        current1 = current1->next;
    }
}

void reverseList(Node** head) {
    Node* prev = NULL;
    Node* current = *head;
    Node* nextNode;

    while (current != NULL) {
        nextNode = current->next;
        current->next = prev;
        prev = current;
        current = nextNode;
    }

    *head = prev;
}
```

```
void concatenateLists(Node** list1, Node* list2) {
    if (*list1 == NULL) {
        *list1 = list2;
    } else {
        Node* current = *list1;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = list2;
    }
}

int main() {
    Node* list1 = NULL;
    Node* list2 = NULL;

    append(&list1, 3);
    append(&list1, 1);
    append(&list1, 4);

    append(&list2, 2);
    append(&list2, 5);

    printf("Original List 1:\n");
    display(list1);

    printf("\nSorting List 1:\n");
    sortList(&list1);
    display(list1);

    printf("\nReversing List 1:\n");
    reverseList(&list1);
    display(list1);

    printf("\nOriginal List 2:\n");
    display(list2);

    printf("\nConcatenating List 1 and List 2:\n");
    concatenateLists(&list1, list2);
    display(list1);

    return 0;
}
```

Output:



```
Original List 1:
3 -> 1 -> 4 -> NULL

Sorting List 1:
1 -> 3 -> 4 -> NULL

Reversing List 1:
4 -> 3 -> 1 -> NULL

Original List 2:
2 -> 5 -> NULL

Concatenating List 1 and List 2:
4 -> 3 -> 1 -> 2 -> 5 -> NULL

Process returned 0 (0x0)   execution time : 0.033 s
Press any key to continue.
```

2. 8)WAP to Implement doubly link list with primitive operations

I.Create a doubly linked list.
II. Insert a new node to the left of the node.
III. Delete the node based on a specific value
IV. Display the contents of the list

Code:

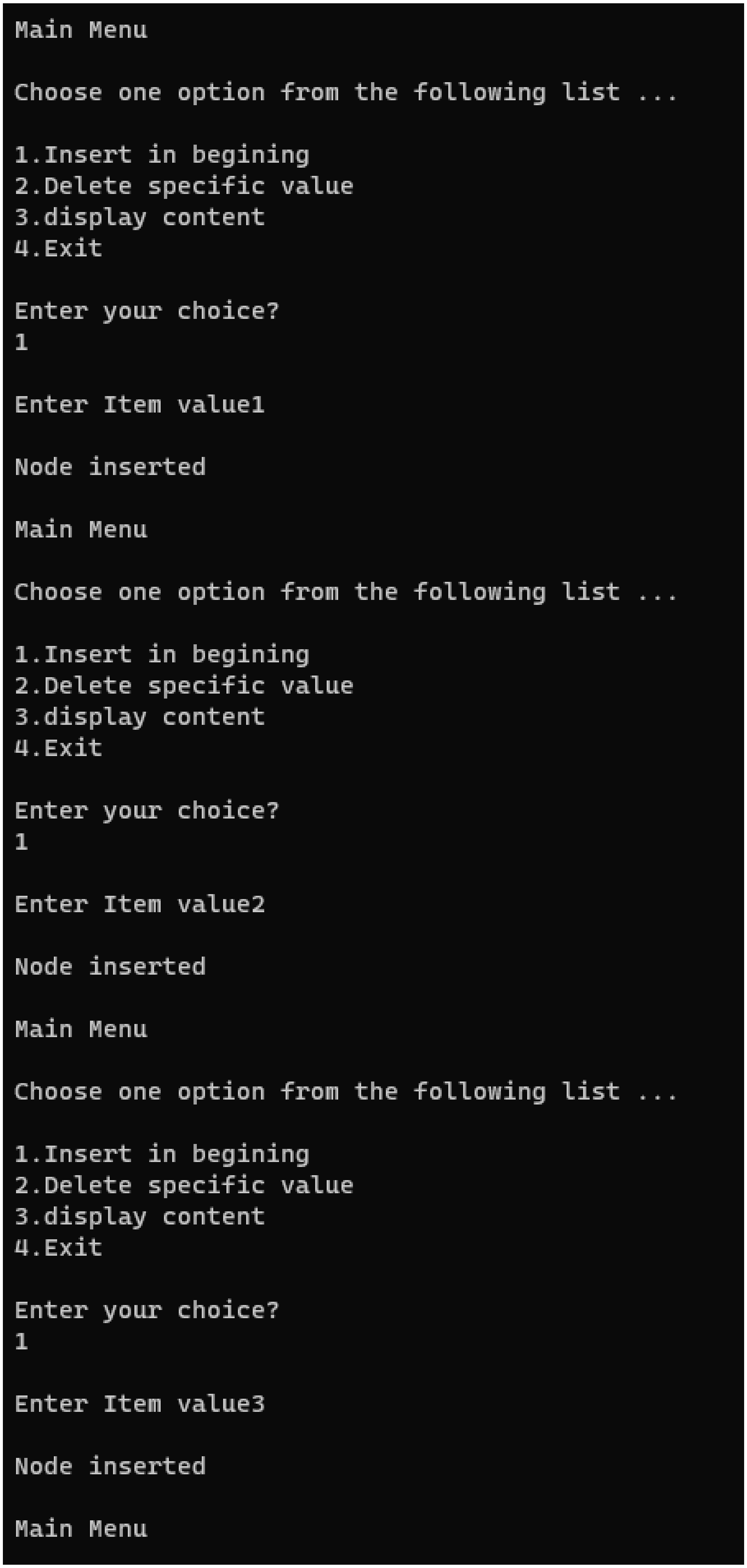
```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   struct node *prev;
   struct node *next;
   int data;
};
struct node *head;
void insertion_beginning();
void deletion_specified();
void display();

void main ()
{
int choice =0;
```

```
    while(choice != 9)
    {
        printf("\nMain Menu\n");
        printf("\nChoose one option from the following list ...\n");
        printf("\n1.Insert in begining\n2.Delete specific value\n3.display content\n4.Exit\n");
        printf("\nEnter your choice?\n");
        scanf("\n%d",&choice);
        switch(choice)
        {
            case 1:
            insertion_beginning();
            break;
            case 2:
            deletion_specified();
            break;
            case 3:
            display();
            break;
            case 4:
            exit(0);
            break;
            default:
            printf("Please enter valid choice..");
        }
    }
}
void insertion_beginning()
{
   struct node *ptr;
   int item;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\nOVERFLOW");
   }
   else
   {
    printf("\nEnter Item value");
    scanf("%d",&item);

   if(head==NULL)
   {
      //creating a list
       ptr->next = NULL;
```

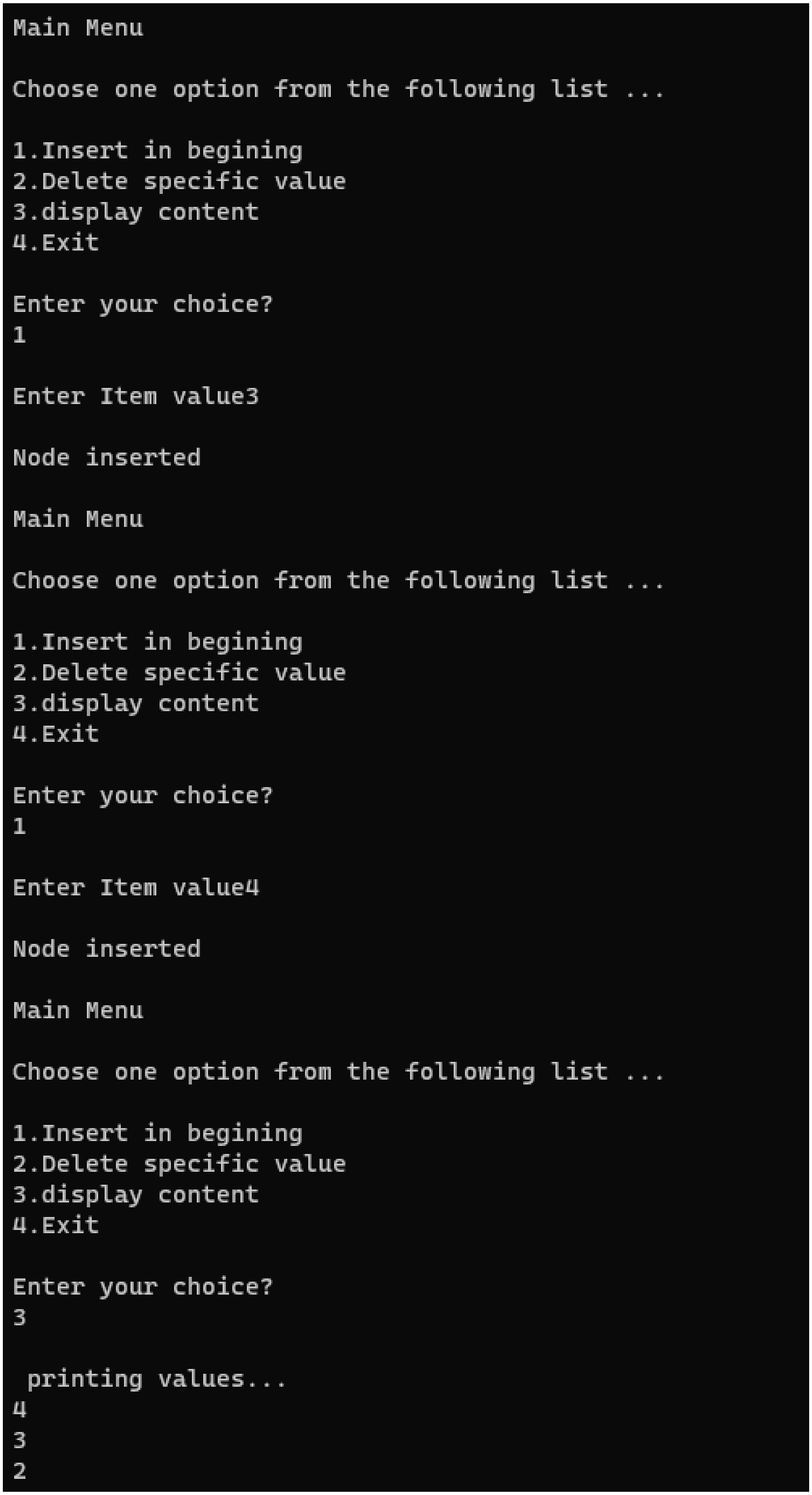
```
        ptr->prev=NULL;
        ptr->data=item;
        head=ptr;
    }
    else
    {
        ptr->data=item;
        ptr->prev=NULL;
        ptr->next = head;
        head->prev=ptr;
        head=ptr;
    }
    printf("\nNode inserted\n");
}

}

void deletion_specified()
{
     struct node *ptr, *temp;
     int val;
     printf("\n Enter the data after which the node is to be deleted : ");
     scanf("%d", &val);
     ptr = head;
     while(ptr -> data != val)
     ptr = ptr -> next;
     if(ptr -> next == NULL)
     {
         printf("\nCan't delete\n");
     }
     else if(ptr -> next -> next == NULL)
     {
         ptr ->next = NULL;
     }
     else
     {
         temp = ptr -> next;
         ptr -> next = temp -> next;
         temp -> next -> prev = ptr;
         free(temp);
         printf("\nnode deleted\n");
     }
}
void display()
```

```
{
    struct node *ptr;
    printf("\n printing values...\n");
    ptr = head;
    while(ptr != NULL)
    {
        printf("%d\n",ptr->data);
        ptr=ptr->next;
    }
}
```

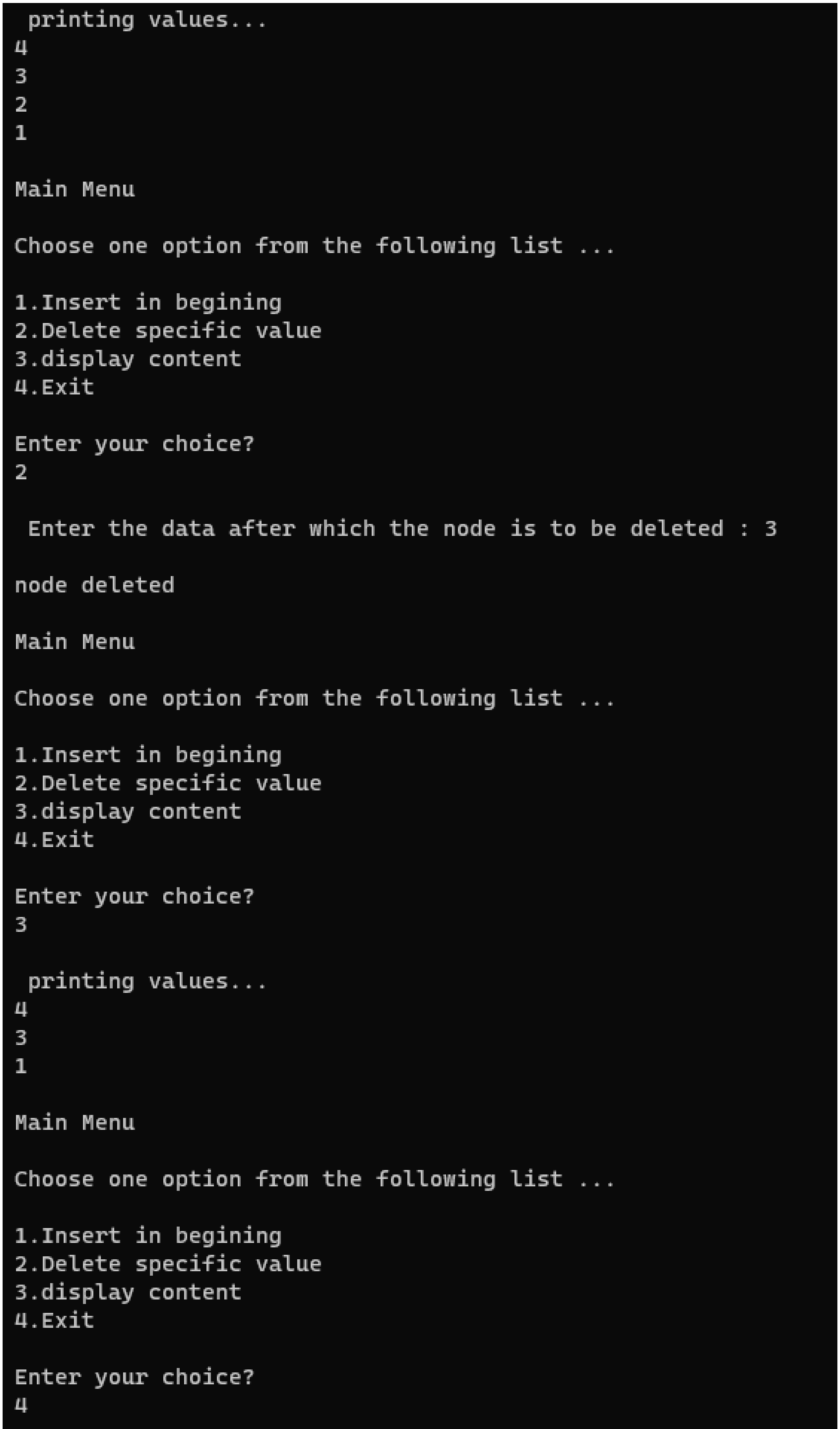
Output:



```
Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
1

Enter Item value1

Node inserted

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
1

Enter Item value2

Node inserted

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
1

Enter Item value3

Node inserted

Main Menu
```



```
Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
1

Enter Item value3

Node inserted

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
1

Enter Item value4

Node inserted

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
3

 printing values...
4
3
2
```



```
 printing values...
4
3
2
1

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
2

 Enter the data after which the node is to be deleted : 3

node deleted

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
3

 printing values...
4
3
1

Main Menu

Choose one option from the following list ...

1.Insert in begining
2.Delete specific value
3.display content
4.Exit

Enter your choice?
4
```