DS Assignment 8

Implement the following operations in context to singly linked list:

1) Creation

2) Insertion (at begining, middle and end)

3) Deletion (from begining, middle and end)

Source Code:

```
#include<stdio.h>

#include<stdlib.h>

struct node{

        int data;

        struct node* next;

};

struct node *head,*temp;

int main()

{

   int n,choice=1,pos,nom=1;

   printf("Enter Your
Choice:\n1)Creation\n2)Insertion\n3)Deletion\n4)Exit\n");

   while(1)

   {
```

```
    printf("\nEnter your choice among Preferences: ");
        scanf("%d", &n);
        switch(n){
            case 1:
                head=NULL;
                while(choice){
                    struct node* newnode=(struct
node*)malloc(sizeof(struct node));
                    printf("Enter the Value to be Inserted: ");
                    scanf("%d", &newnode->data);
                    newnode->next=NULL;
                    if(head==NULL)
                    {
                        head=temp=newnode;
                    }
                    else
                    {
                        temp->next=newnode;
                        temp=newnode;
                    }
                    nom++;
                    printf("If want to insert More Enter 1 else to
Display enter 0:  ");
```

```
                scanf("%d", &choice);
                }
                struct node* temp=head;  //Display
                while(temp!=NULL)
                {
                        printf("%d ", temp->data);
                        temp=temp->next;
                }
                break;
        case 2:
                printf("Enter the Node Position: ");
                scanf("%d", &pos);
                struct node* newnode=(struct
node*)malloc(sizeof(struct node));
                printf("Enter the Number which you want to
Insert: ");
                scanf("%d", &newnode->data);
                if(pos==1)   //Insertion at Beginning
                {
                    newnode->next=head;
                    head=newnode;
                    nom++;
                }
```

```
  else if(pos==nom) //Insertion at end
  {
        newnode->next=NULL;
        temp=head;
while(temp->next!=NULL)
        {
              temp=temp->next;
        }
        nom++;
        temp->next=newnode;
  }
  else //Insertion in Nth position
  {
        int i=1;
        temp=head;
        while(i<pos-1)
        {
              temp=temp->next;
              i++;
        }
        nom++;
   newnode->next=temp->next;
```

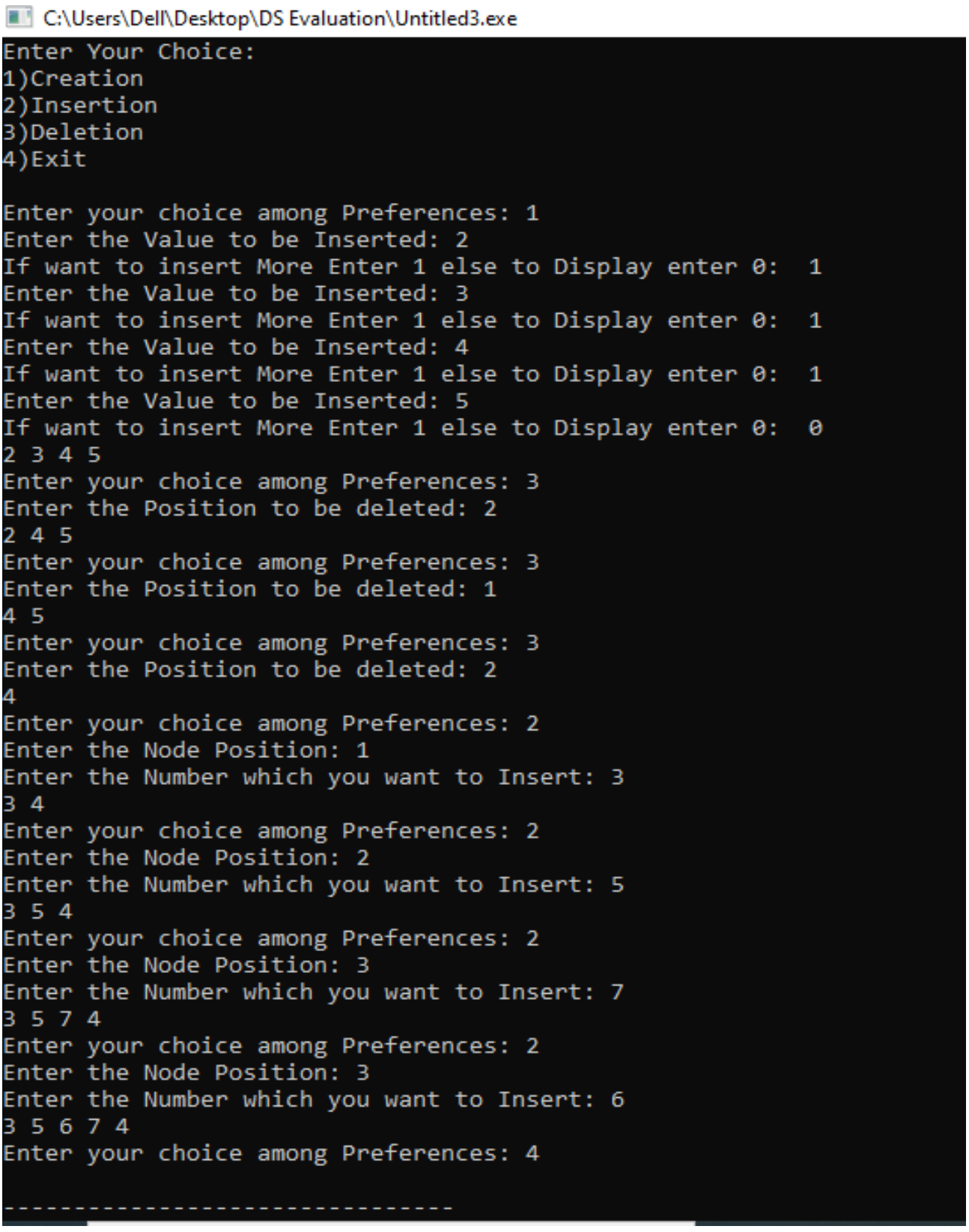
```
            temp->next=newnode;
          }
        temp=head;   //Display
    while(temp!=NULL)
        {
            printf("%d ", temp->data);
            temp=temp->next;
        }
        break;
case 3:
        if(head==NULL)
        {
            printf("The list is empty!");
        }
        else
        {
           int dpos;
            printf("Enter the Position to be deleted: ");
            scanf("%d", &dpos);
            if(dpos==1) //Deletion from Beginning
            {
              temp=head;
```

```
                head=head->next;
                free(temp); //Remove the deleted node
from memory
                nom--;
        }
        else if(dpos==nom-1) //Deletion from End
        {
                struct node* prevnode;
                temp=head;
                while(temp->next!=NULL)
                {
                        prevnode=temp;
                        temp=temp->next;
                }
                if(temp==head)
                {
                        head=NULL;
                        free(temp);
                }
                else
                {
                        prevnode->next=NULL;
```

```
                free(temp);
            }
            nom--;
        }
        else
        {
            int i=1;
            struct node* newnode;
            temp=head;
            while(i<pos-1)
            {
                temp=temp->next;
                i++;
            }
            newnode=temp->next;
            temp->next=newnode->next;
free(newnode);
nom--;
        }
        temp=head;   //Display
    while(temp!=NULL)
        {
```

```
                    printf("%d ", temp->data);
                    temp=temp->next;
                }
                break;
            }
        case 4:
            exit(0);
        default:
            printf("Enter a valid Number among all\n");
            break;
    }
    }
}
```

Output:



Program ends at 4