IBM19CS044 link list insertion

Deepu K

```
#include<stdio.h>
#include<stdlib.h>
Void create();
Void display();
Void insert_at_Beginning();
Void insert_at_End();
Void insert_at_middle();
Struct node
{
  int data;
  Struct node *next;
};

Struct node *head=NULL;
int main(int argc, char *argv)
{
  int choice, ele;
  Char ch;
```

```
do
{
Printf("1. create \n 2. Display \ 3. Insert_at_
        beginning \n 4. Insert_at_end \n 5. Insert_at_ middle");
Printf(" Enter your choice : ");
 Scanf("%d", & choice);
switch(choice) {
    case 1: create(); break;
    Case 2: display(); break;
     Case 3: insert_at-Beginning(); break;
    Case 4: insert-at-End; break;
    Case 5: insert-at-Middle; break;
  }
 } while (choice != 6);
}

Void create()
{
  Struct node *newnode, *temp;
  int item;

newnode = (Struct node*) malloc
            (Sizeof(Struct node));
```

```
printf(" Enter the data : ");
 scanf(" %d", &item);
 newnode -> data = item;
 if (head == NULL)
 {
 newnode -> next = NULL;
  head = newnode;
  printf(" node created \n");
}
else
{
temp = head;
while (temp -> next != NULL)
{
   temp = temp -> next;
}
 temp -> next = newnode;
 newnode -> next = NULL;
 printf(" node created \n");
 }
}
```

```
Void display()
{
  Struct node *ptr = NULL;
    ptr = head
  if (ptr == NULL)
  {
   Printf(" Nothing to Print);
  }
  else
  {
   while (ptr != NULL)
   {
   Printf("%d", ptr -> data);
    ptr = ptr -> next;
    }
  }
}

Void insert_at_Beginning()
{
  Struct node *newnode;
  int ele;
```

```
Printf("Enter the element : ");
Scanf("%d", &ele);

newnode = (struct node*) malloc
            (sizeof(struct node));
if(head == NULL)
{
  Printf("empty list");
} else {
  newnode -> data = ele;
  newnode -> next = head;
   head = newnode;
  }
}

Void insert-at-End()
{
  Struct node *newnode, *temp;
  int ele;
  printf("enter the element : ");
   Scanf("%d", &ele);
newnode = (struct node*) malloc
            (sizeof(struct node));
```

```
if (head == NULL)
{
    printf("empty list");
}
else {
newnode -> data = ele;
newnode -> next = NULL;
  temp = head;
while( temp != NULL && temp->next != NULL)
  temp = temp -> next;
  temp -> next = newnode;
  }
}

Void  insert_at_middle()
{
  Struct node *newnode, *temp;
  int ele, position;

Printf("enter the element : ");
  Scanf("%d", &ele);
Printf("enter the position to be inserted :");
  Scanf("%d", &Position);
```

```
newnode = (struct node*) malloc
            (sizeof(struct node));
if (head == NULL)
printf(empty list);
else {
 newnode -> data = ele;
 newnode -> next = NULL;
  temp = head
for(int i=2; i<position-1; i++)
{
temp = temp -> next;
  if (temp == NULL)
  break;
  3 if(temp != NULL)
   newnode -> next = temp -> next;
   temp -> next = newnode;
   3
  3
 3
```