C Program To Implement Linked List Using Array Abstract Data Type

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

void create( )
void insert( )
void delete( )
void display( )

struct node
{
int data;
struct node *link;
};
struct node *first = NULL, *last = NULL, *next, *prev, *cur;

void main( )
{
int ch;
clrscr( );
printf ("\n SINGLY LINKED LIST");
do
{
printf ("\n 1.CREATE \n 2.INSERT\n 3.DELETE \n 4.EXIT \n");
printf ("\n ENTER YOUR CHOICE: ");
scanf ("%d", &ch );
switch (ch)
{
case 1:
create( );
display( );
break;
case 2:
insert( );
display( );
break;
case 3:
delete( );
display( );
break;
case 4:
exit(0);
}
} while( ch <= 3)
}
void create( )
{
cur = ( struct node*)malloc (sizeof (struct node));
printf ("\n ENTER THE DATA: ");
```

```
scanf ("%d" , &cur?data);
cur?link = NULL;
first = cur;
last = cur;
}
void insert( )
{
int pos, c = 1;
cur = (struct node *)malloc(sizeof (struct node) );
printf ("\ENTER THE DATA :");
scanf ("%d", &cur?data);
printf("\ENTER THE POSITION:");
scanf ("%d", &pos );
if ( (pos = = 1)&& (first ! = NULL) )
{
cur?link = first;
first = cur;
}
else
{
next = first;
while (c< pos )
{
prev = next;
next = prev?link;
c++;
}
if ( prev = = NULL)
{
printf ("\n INVALID POSITION \n");
}
else
{
cur?link = prev?link;
prev?link = cur;
if (cur?link = = NULL)
{
last = cur;
}
}
}
}

void delete( )
{
int pos, c=1;
printf ("\ENTER THE POSITION :");
scanf ("%d", &pos);
if (first = = NULL)
{
printf ("\n LIST IS EMPTY \n");
}
else if (pos = = 1&& first?link == NULL)
{
printf(“\Ndeleted element is %d\n”, cur?data);
free(cur);
}
else
{
next = first;
while (c< pos )
{
prev = next ;
next = next?link;
c++;
}
prev?link = next ?link;
next?link = NULL;
if (next = = NULL)
{
printf (“\n INVALID POSITION \n “);
}
else
{
printf (“\n DELETED ELEMENT IS%d\n”, next?data);
free (next);
if(prev?link == NULL)
{
last = prev;
}
}
}
}

void display( )
{
cur = first;
while (cur! = NULL)
{
printf (“\n%d”, cur?data);
cur = cur?link;
}
}
```