Singly Linked List

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

void add\_beg();

void delete\_();

void display();

void add\_bet();

void del\_bet();

void add\_end();

void del\_end();

struct node

{

int data;

struct node \*link;

}\*LIST=NULL,\*p,\*q;

void main()

{

int ch;

clrscr();

do

{

printf("1.Add\_Beg 2.Del\_Beg 3.Display 4.Exit 5.Add\_Bet 6.Del\_Bet 7.Add\_End 8.Del\_End\n");

printf("ENTER THE CHOICE:");

scanf("%d",&ch);

switch(ch)

{

case 1:

add\_beg();

break;

case 2:

delete\_();

break;

case 3:

display();

break;

case 5:

add\_bet();

break;

case 6:

del\_bet();

break;

case 7:

add\_end();

break;

case 8:

del\_end();

break;

}

}while(ch!=4);

getch();

}

void add\_beg()

{

int a;

q=malloc(sizeof(struct node));

printf("Enter Data :");

scanf("%d",&a);

q->data=a;

if(LIST==NULL)

{

LIST=q;

q->link=NULL;

}

else

{

q->link=LIST;

LIST=q;

}

}

void delete\_()

{

if(LIST==NULL)

{

printf("UNDERFLOW");

}

else

{

q=LIST;

LIST=LIST->link;

free(q);

}

}

void display()

{

p=LIST;

while(p)

{

printf("%d-->",p->data);

p=p->link;

}

printf("\n");

}

void add\_bet()

{

int pos;

int a,i,t;

q=malloc(sizeof(struct node));

printf("Enter Data:");

scanf("%d",&a);

q->data=a;

printf("Enter Position :");

scanf("%d",&pos);

p=LIST;

for(i=0;i<pos-1;i++)

{

p=p->link;

}

q->link=p->link;

p->link=q;

}

void del\_bet()

{

int pos,i;

printf("Enter the position for deleting the node:");

scanf("%d",&pos);

p=LIST;

for(i=0;i<pos-1;i++)

{

p=p->link;

}

printf("%d is deleted",p->link->data);

p->link=p->link->link;

}

void add\_end()

{

q=(struct node\*)malloc (sizeof(struct node));

printf("Enter the data:");

scanf("%d",&q->data);

q->link=NULL;

if(LIST==NULL)

{

LIST=q;

}

else

{

p=LIST;

while(p->link!=NULL)

{

p=p->link;

}

p->link=q;

}

}

void del\_end()

{

int temp;

if(LIST==NULL)

{

printf("No item");

}

else

{

if(LIST->link!=NULL)

{

p=LIST;

while(p->link->link!=NULL)

{

p=p->link;

}

printf("%d deleted\n",p->link->data);

p->link=NULL;

free(p->link->link);

}

else if(LIST->link==NULL)

{

printf("%d deleted\n",LIST->data);

LIST=NULL;

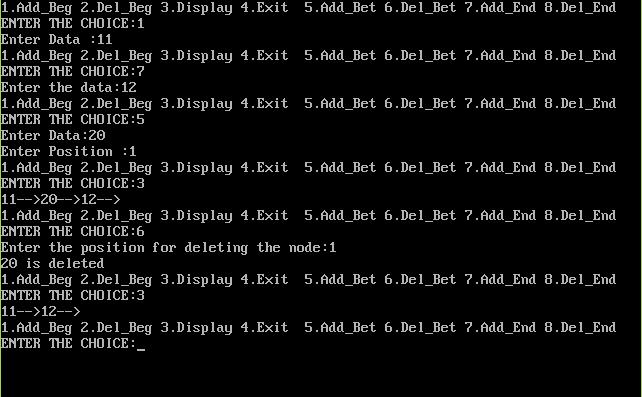
free(LIST);

}

}

}

OUTPUT:



Reversing A List

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

static void reverse(struct node\*\* head\_ref);

void add\_beg();

struct node

{

int data;

struct node \*link;

}\*LIST=NULL,\*p,\*q;

static void reverse(struct node\*\* head\_ref)

{

struct node\* prev = NULL;

struct node\* current = \*head\_ref;

struct node\* next;

while (current != NULL)

{

next = current->link;

current->link = prev;

prev = current;

current = next;

}

\*head\_ref = prev;

}

void add\_beg()

{

int a;

q=malloc(sizeof(struct node));

printf("Enter Data :");

scanf("%d",&a);

q->data=a;

if(LIST==NULL)

{

LIST=q;

q->link=NULL;

}

else

{

q->link=LIST;

LIST=q;

}

}

void display()

{

p=LIST;

while(p)

{

printf("%d-->",p->data);

p=p->link;

}

printf("\n");

}

void main()

{

int ch;

clrscr();

do

{

printf("1.Add\_Beg 2.Reverse 3.Display 4.Exit \n");

printf("ENTER THE CHOICE:");

scanf("%d",&ch);

switch(ch)

{

case 1:

add\_beg();

break;

case 2:

reverse(&LIST);

break;

case 3:

display();

break;

}

}while(ch!=4);

getch();

}

OUTPUT:

