/*stack using linked list*/

```
#include <stdio.h>
#include <stdlib.h>
struct stack
{
int data;
struct stack*link;
};
struct stack*top;                    //here,stack_head=top
struct stack*push(struct stack*,int);
struct stack*pop(struct stack*);
struct stack*display(struct stack*);
int main()
{
int item, ch;
while (1)
{
printf("**main menu**\n");
printf("1 - Push\n");
printf("2 - Pop\n");
printf("3 - Dipslay\n");
printf("4 - Exit\n");
printf("Enter your choice : \n");
scanf("%d", &ch);
switch (ch)
{
case 1:
printf("Enter the no to be pushed on stack : ");
scanf("%d", &item);
top=push(top,item);
break;
```

```
case 2:
top=pop(top);       //return the rest of list after popping out a data
break;
case 3:
top=display(top);
break;
case 4:
exit(0);
default :
printf(" invalid choice\n");
}
}
}
/* Push data into stack */
struct stack*push(struct stack*top,int item)
{
struct stack*new_node;
new_node=(struct stack*)malloc(sizeof(struct stack*));
new_node->data=item;
if (top==NULL)
{
new_node->link=NULL;
top=new_node;
}
else
{
new_node->link=top;
top=new_node;
}
printf("the item is pushed\n");
return top;
```

```
}
/* Pop Operation on stack */
struct stack*pop(struct stack*top)
{
struct stack*ptr;
if (top==NULL)
{
printf("stack is empty\n");
}
else
{
ptr=top;
top=top->link;
printf("the popped value : %d\n",ptr->data);
free(ptr);
}
return top;
}
/* Display stack elements */
struct stack*display(struct stack*top)
{
struct stack*ptr;
if (top==NULL)
{
printf("stack is empty\n");
}
else
{
printf("the stack is below\n");
ptr=top;
while (ptr!=NULL)
```

```
{
printf("%d\n",ptr->data);
ptr=ptr->link;
}
}
return top;
}
```