**C-Program to implement STACK using LINKED LIST**

#include<stdio.h>

#include<conio.h>

#include<alloc.h>

#include<process.h>

struct node

{

int info;

struct node \*link;

};

typedef struct node \*NODE;

NODE getnode()

{

NODE x;

x=(NODE)malloc(sizeof(struct node));

if(x==NULL)

{

printf("mem full\n");

exit(0);

}

return x;

}

void freenode(NODE x)

{

free(x);

}

NODE insert\_front(NODE first,int item)

{

NODE temp;

temp=getnode();

temp->info=item;

temp->link=NULL;

if(first==NULL)

return temp;

temp->link=first;

first=temp;

return first;

}

NODE delete\_front(NODE first)

{

NODE temp;

if(first==NULL)

{

printf("stack is empty cannot delete\n");

return first;

}

temp=first;

temp=temp->link;

printf("item deleted at front-end is=%d\n",first->info);

free(first);

return temp;

}

void display(NODE first)

{

NODE temp;

if(first==NULL)

printf("stack empty cannot display items\n");

for(temp=first;temp!=NULL;temp=temp->link)

{

printf("%d\n",temp->info);

}

}

void main()

{

int item,choice,pos;

NODE first=NULL;

clrscr();

for(;;)

{

printf("\n 1:Insert\_front\n 2:Delete\_front\n 3:Display\_list\n 4:Exit\n");

printf("enter the choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("enter the item at front-end\n");

scanf("%d",&item);

first=insert\_front(first,item);

break;

case 2:first=delete\_front(first);

break;

case 3:display(first);

break;

default:exit(0);

break;

}

}