

**19BIT0292**

**Bhaumik Tandan**

**ASSESSMENT-1**

**DATA STRUCTURES**

**AND**

**ALGORITHMS**

**LABORATORY**

CSE2011

L57+L58

**Q1)** Implement Stack and realize various operations to be carried out on it.

**stack.h**

**CODE**

#include "./stack\_header/varibles\_decalred.h"//it also contains header files

#include "./stack\_header/stack\_functions.h"

#include "./stack\_header/push\_type.h"

#pragma once//restrict double import

#define push(st,a) \_Generic(a, int: pushi\_\_19BIT0292, char\*: pushs\_\_19BIT0292,double: pushf\_\_19BIT0292,char:pushc\_\_19BIT0292,float:pushf\_\_19BIT0292)(st,a)//char and int will be treated similarly

void s\_in(stack \*s)

{

s->t\_\_19BIT0292=-1;

s->stack\_\_19BIT0292=0;

s->d\_type\_\_19BIT0292=0;

}

void menu(stack \*st)

{

void\* (\* fp[3])(stack \*);

//0 push

//1 pop

//2 top

//3 display whole stack

fp[0]=&pop;

fp[1]=&top;

fp[2]=&display;

printf("\n\n\n1)Push\n2)Pop\n3)Top\n4)Display\n5)Exit\n");

printf("\nEnter your choice: ");

int c;

scanf("%d",&c);

if(c==1)

{

printf("\n\nEnter that you to push in the stack: ");

char s[21];//this will get destroyed after function is finished it also has null

scanf("%s",s);

int a=atoi(s);//convert string to int

float f=atof(s);

if((a!=0 || strcmp("0",s)==0)&& f==a)

{

push(st,a);

return menu(st);

}

if(f!=0)

{

push(st,f);

return menu(st);

}

if(strlen(s)>1){

push(st,s);

}

else

push(st,s[0]);

return menu(st);

}

else if(c==5)

return;

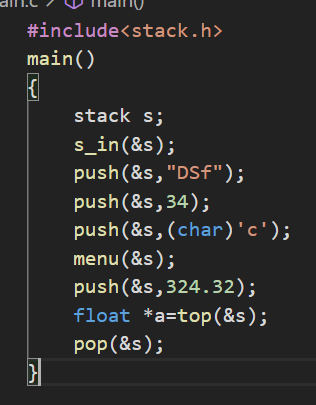
fp[c-2](st);

return menu(st);

}

**main.c**

**CODE**

#include<stack.h>

main()

{

stack s;

s\_in(&s);

push(&s,"DSf");

push(&s,34);

push(&s,(char)'c');

menu(&s);

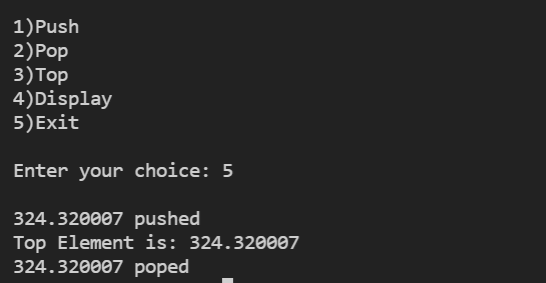
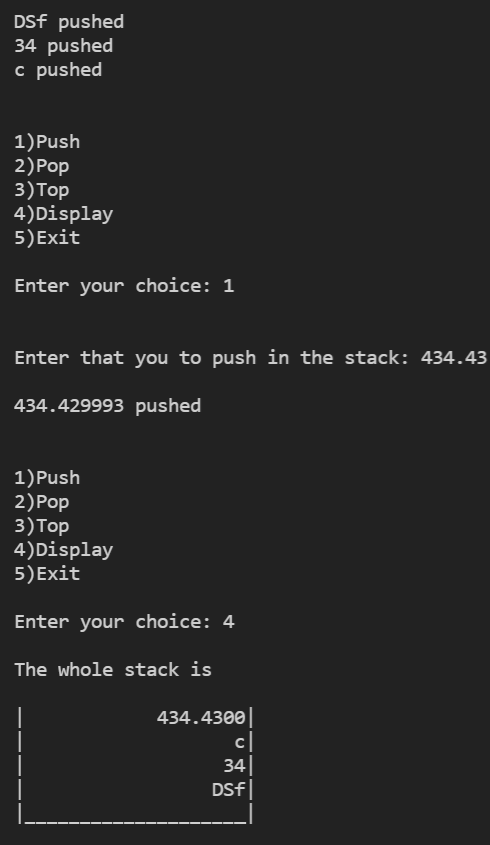
push(&s,324.32);

float \*a=top(&s);

pop(&s);

}

**OUTPUT**

****

# [**CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE**](https://github.com/Bhaumik-Tandan/Generic_Stack_using_c)

**Q2)** Implement Queue and realize various operations to be carried out on it.

**queue.h**

**CODE**

#include "./queue\_header/varibles\_decalred.h"//it also contains header files

#include "./queue\_header/queue\_functions.h"

#include "./queue\_header/enqueue\_type.h"

#pragma once//restrict double import

#define enqueue(s,a) \_Generic(a, int: enqueuei\_\_19BIT0292, char\*: enqueues\_\_19BIT0292,double: enqueuef\_\_19BIT0292,char:enqueuec\_\_19BIT0292,float:enqueuef\_\_19BIT0292)(s,a)

void q\_in(queue \*q)

{

q->r\_\_19BIT0292=-1;

q->queue\_\_19BIT0292=0;

q->d\_type\_\_19BIT0292=0;

}

void menu(queue \*q)

{

void\* (\* fp[4])(queue\*);

fp[0]=&denqueue;

fp[1]=&front;

fp[2]=&rear;

fp[3]=&display;

printf("\n\n\n1)Enqueue\n2)Dequeue\n3)Front\n4)Rear\n5)Display\n6)Exit\n");

printf("\nEnter your choice: ");

int c;

scanf("%d",&c);

if(c==1)

{

printf("\n\nEnter that you to enqueue in the stack: ");

char s[21];

scanf("%s",s);

int a=atoi(s);//convert string to int

float f=atof(s);

if((a!=0 || strcmp("0",s)==0)&& f==a)

{

enqueue(q,a);

return menu(q);

}

if(f!=0)

{

enqueue(q,f);

return menu(q);

}

if(strlen(s)>1)

enqueue(q,s);

else

enqueue(q,s[0]);

return menu(q);

}

else if(c==6)

return;

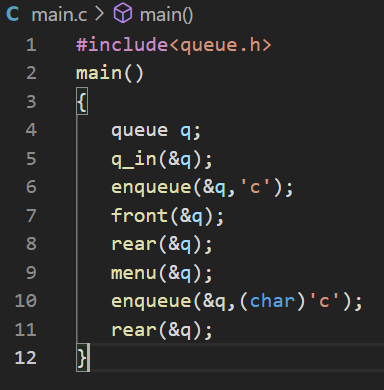
fp[c-2](q);

return menu(q);

}

**main.c**

**CODE**

#include<queue.h>

main()

{

queue q;

q\_in(&q);

enqueue(&q,'c');

front(&q);

rear(&q);

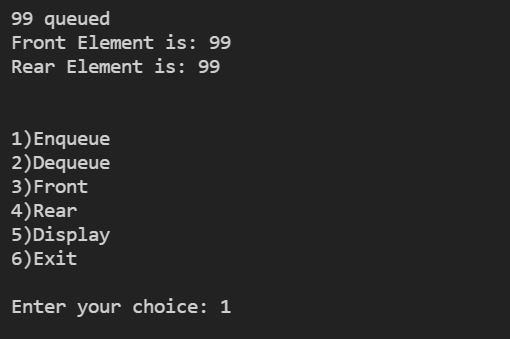
menu(&q);

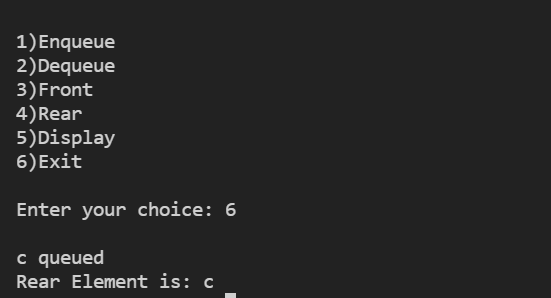
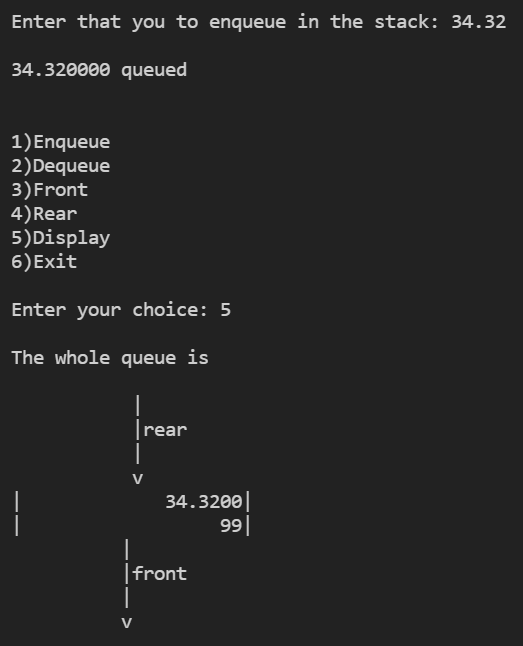
enqueue(&q,(char)'c');

rear(&q);

}

**OUTPUT**



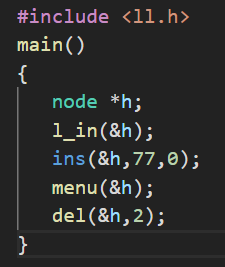


# [**CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE**](https://github.com/Bhaumik-Tandan/Generic_queue_in_c)

**Q3)** Create a linked list and perform various operations to be carried out on the linked list.

**main.c**

**CODE**

#include <ll.h>

main()

{

node \*h;

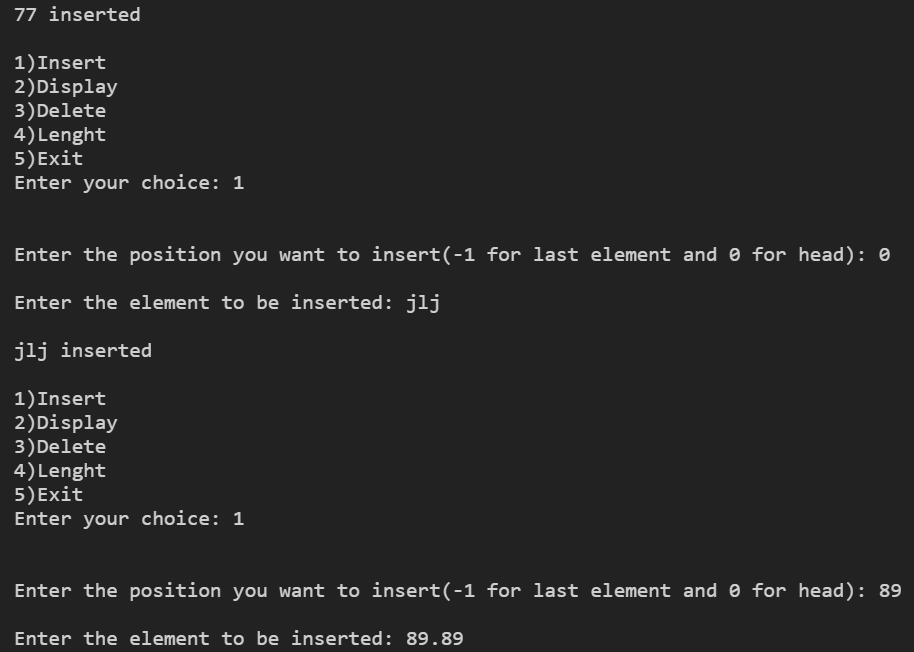
l\_in(&h);

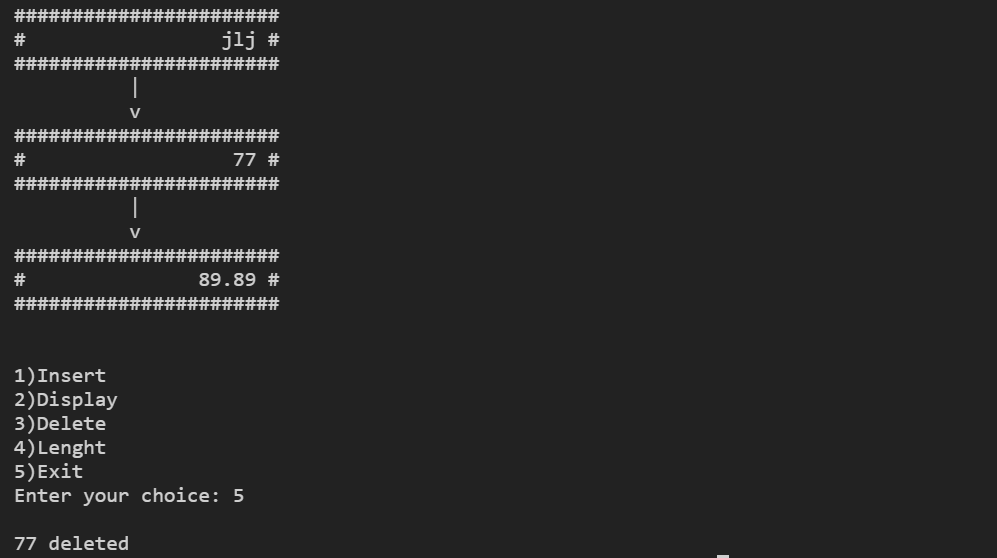
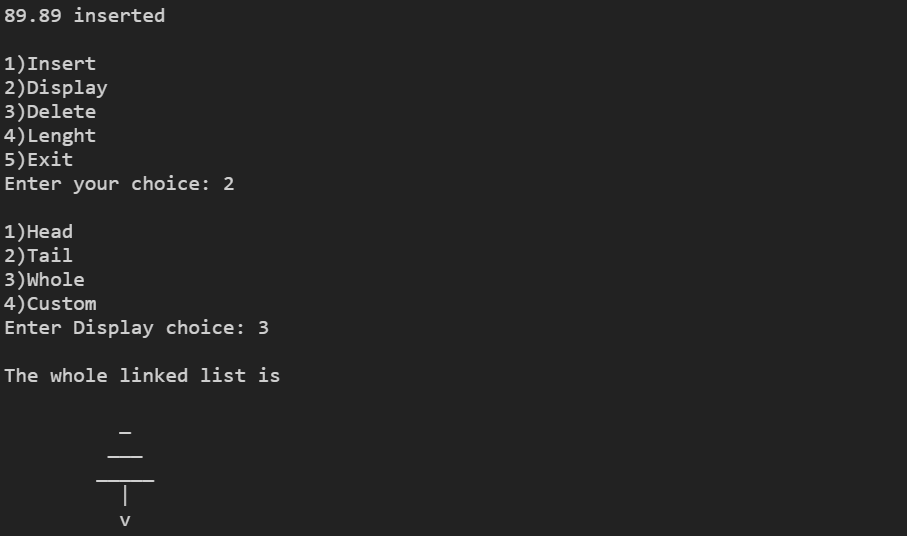
ins(&h,77,0);

menu(&h);

del(&h,2);

}

**OUTPUT**



# [**CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE**](https://github.com/Bhaumik-Tandan/Generic_linked_list_in_c)

**Q4) Multiply two matrices of order ( m X n) and ( n X p ). If the order of the matrices is other than this, then supply an error message.**

**CODE**

#include<stdio.h>

#include<stdlib.h>

void fun()

{

printf("\n\n1)Enter Again\n2)Exit\nEnter your choice: ");

int c;

scanf("%d",&c);

if(c==1)

main();

else

exit(0);

}

main()

{

int\*\*\* m;//stores both the matrix;

m=malloc(sizeof(int\*\*)\*2);

int r[2],c[2];

for(int i=0;i<2;i++){

printf("\nEnter the number of rows in matrix %d: ",i+1);

scanf("%d",r+i);

printf("\nEnter the number of collumns in matrix %d: ",i+1);

scanf("%d",c+i);

}

if(r[1]!=c[0])

{

printf("\nWrong matrix shape");

fun();

}

for(int i=0;i<2;i++){

m[i]=malloc(sizeof(int\*)\*r[i]);

for(int j=0;j<r[i];j++){

m[i][j]=malloc(sizeof(int)\*c[i]);

printf("\nEnter space seperated row %d of matrix %d: ",j+1,i+1);

for(int k=0;k<c[i];k++)

scanf("%d",&m[i][j][k]);

}

}

printf("\nThe result matrix is:-\n\n");

for(int i=0;i<r[0];i++)

{

for(int k=0;k<c[1];k++){

int s=0;

for(int j=0;j<r[1];j++)

s+=m[0][i][j]\*m[1][j][k];

printf("\t%d",s);

}

printf("\n");

}

fun();

}

# 

# 

**OUTPUT**

# 

# 

**CALCULATED USING CALCULATOR**

# 

# [CLICK HERE FOR GITHUB LINK](https://github.com/Bhaumik-Tandan/C_codes/blob/master/matrix_mul.c)

**Q5) Compute the inverse of a square matrix of order ( n X n ). If the matrix is a singular matrix, then supply appropriate error message.**

**main.c**

**CODE**

#include<stdio.h>

#include "matrix\_inv.h"

#include "matrix\_print.h"

main()

{

int r;

printf("\nEnter the number of rows: ");

scanf("%d",&r);

int m[r][r];

for(int j=0;j<r;j++){

printf("\nEnter space seperated row %d: ",j+1);

for(int i=0;i<r;i++)

scanf("%d",&m[j][i]);

}

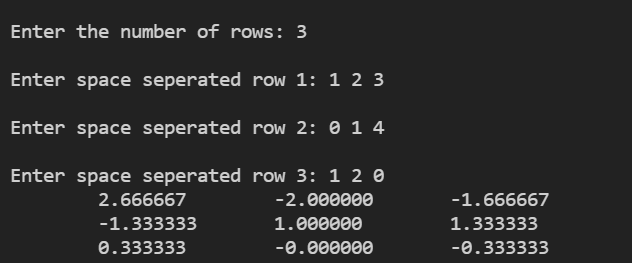
float c[r][r];

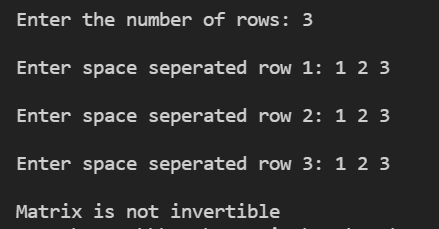
inv(r,m,c);

printff(r,c);

}

**OUTPUT**





# [CLICK HERE FOR GITHUB LINK](https://github.com/Bhaumik-Tandan/C_codes/tree/master/matric_inverse)