VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

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

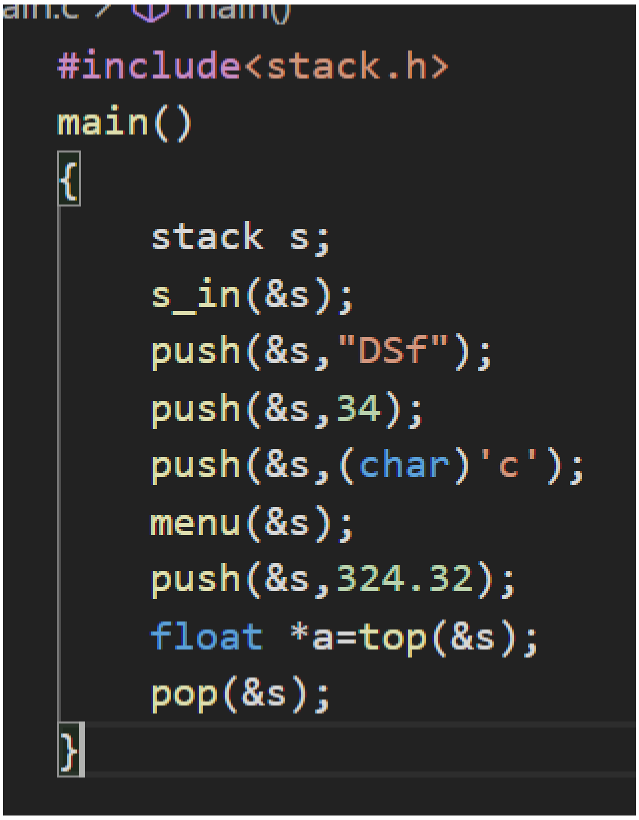
```
#include "./stack_header/varibles_decalred.h"//it also contains header files
#include "./stack_header/stack_functions.h"
#include "./stack_header/push_type.h"
#pragma once//restrict double import
#define push(st,a) _Generic(a, int: pushi__19BIT0292, char*: pushs__19BIT0292,double: pushf__19BIT0292,char:pushc__19BIT0292,float:pushf__19BIT0292)(st,a)//char 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

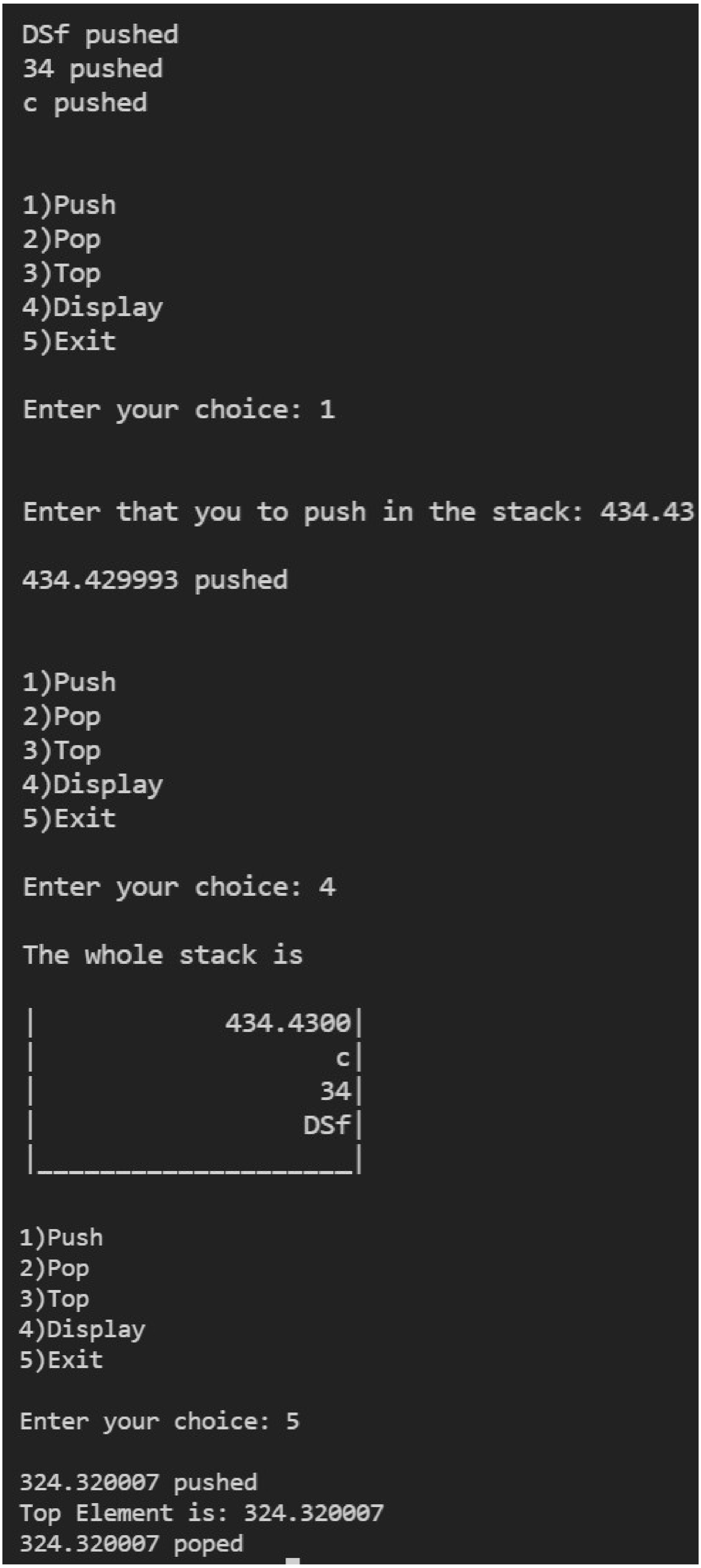
CODE

```
#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);
}
```



```
#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



```
DSf pushed
34 pushed
c pushed


1)Push
2)Pop
3)Top
4)Display
5)Exit

Enter your choice: 1


Enter that you to push in the stack: 434.43

434.429993 pushed


1)Push
2)Pop
3)Top
4)Display
5)Exit

Enter your choice: 4

The whole stack is

|            434.4300|
|                   c|
|                  34|
|                 DSf|
|____________________|

1)Push
2)Pop
3)Top
4)Display
5)Exit

Enter your choice: 5

324.320007 pushed
Top Element is: 324.320007
324.320007 poped
```

CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE

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

queue.h

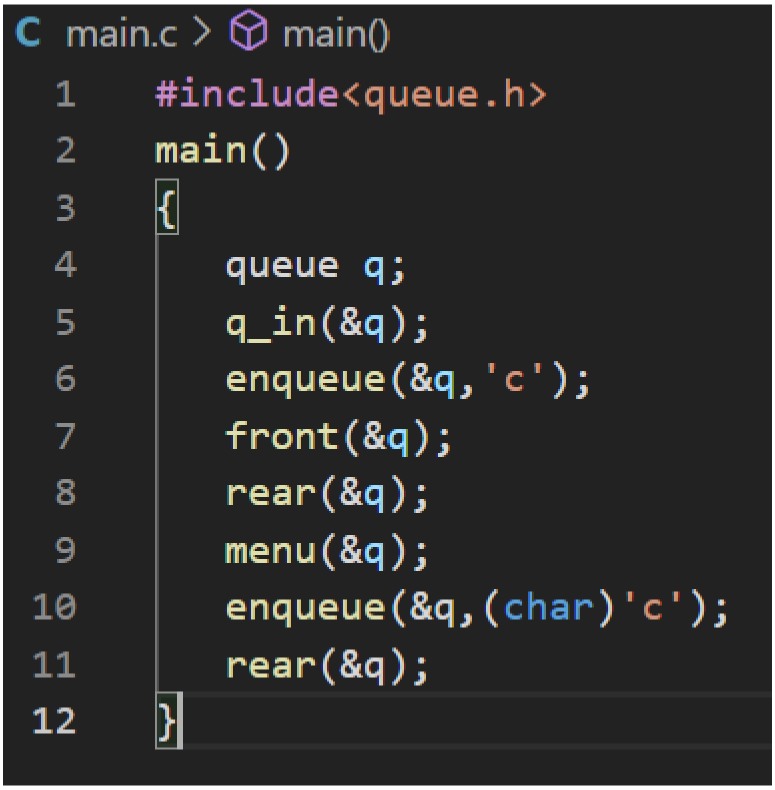
CODE

```
#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

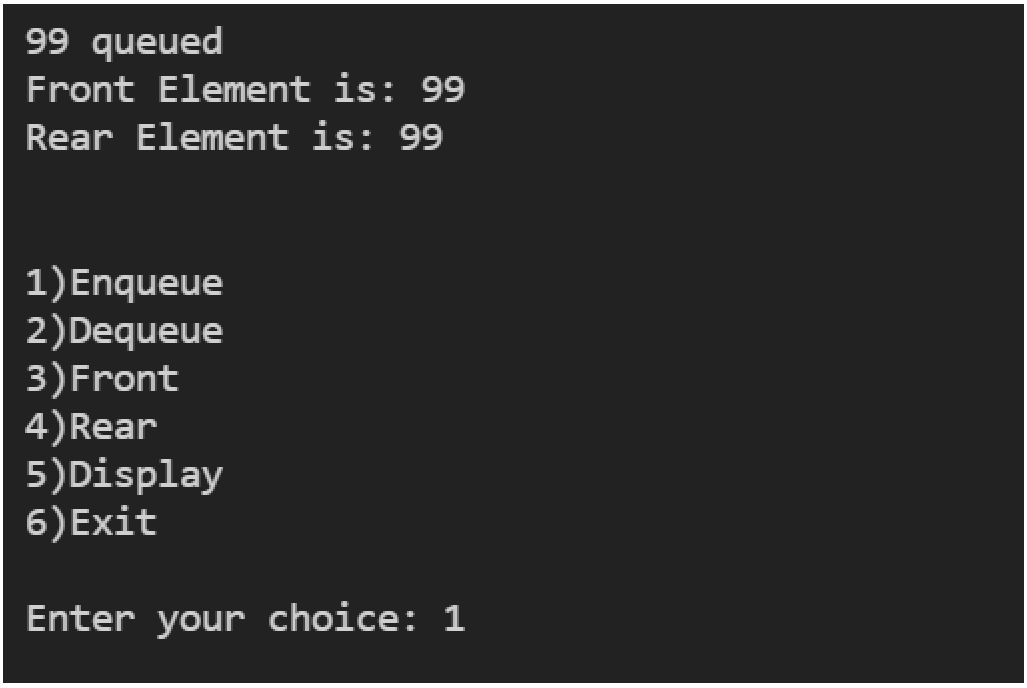
CODE

```c
#include<queue.h>
main()
{
    queue q;
    q_in(&q);
    enqueue(&q,'c');
    front(&q);
    rear(&q);
    menu(&q);
    enqueue(&q,(char)'c');
    rear(&q);
}
```

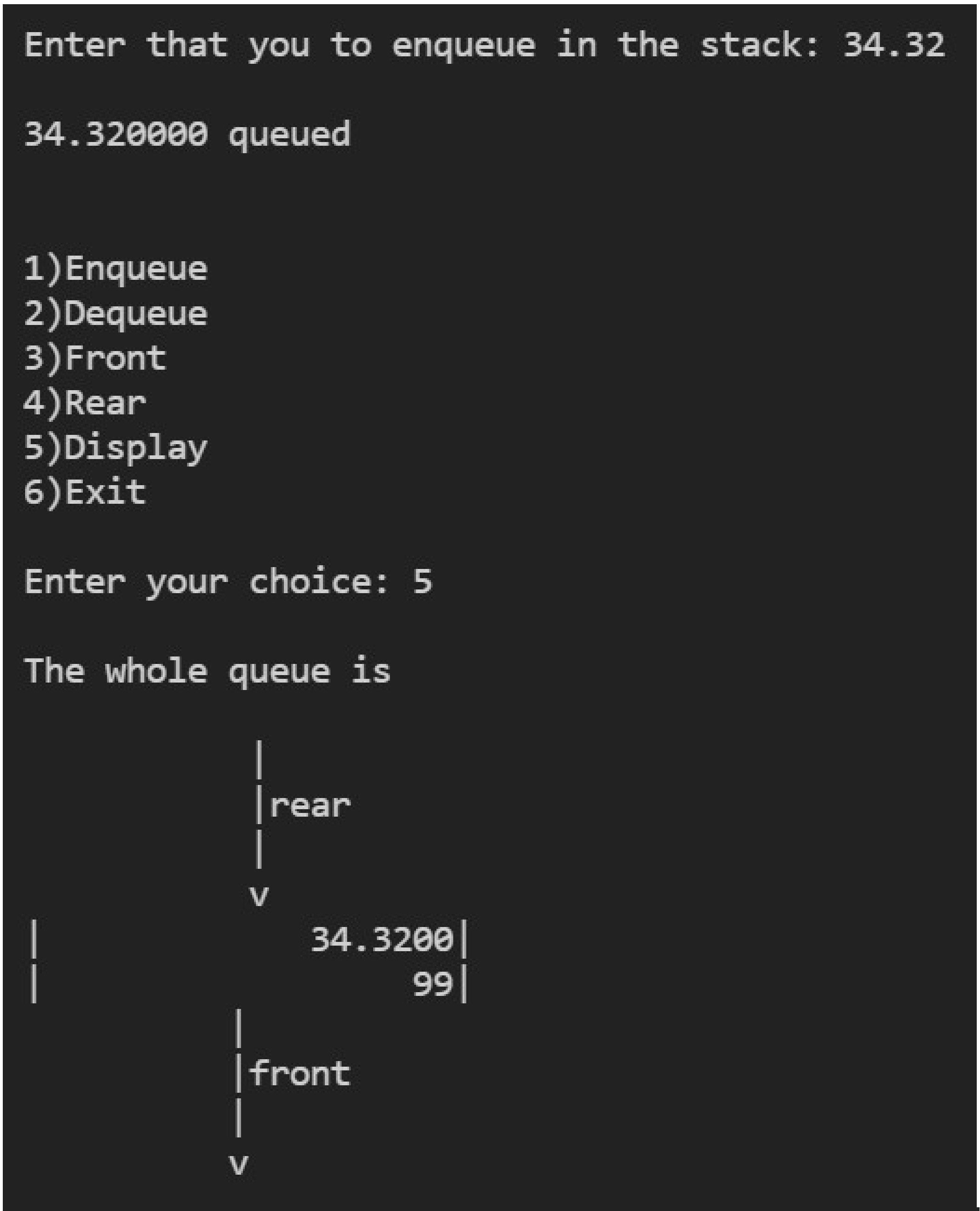


```c
#include<queue.h>
main()
{
    queue q;
    q_in(&q);
    enqueue(&q,'c');
    front(&q);
    rear(&q);
    menu(&q);
    enqueue(&q,(char)'c');
    rear(&q);
}
```

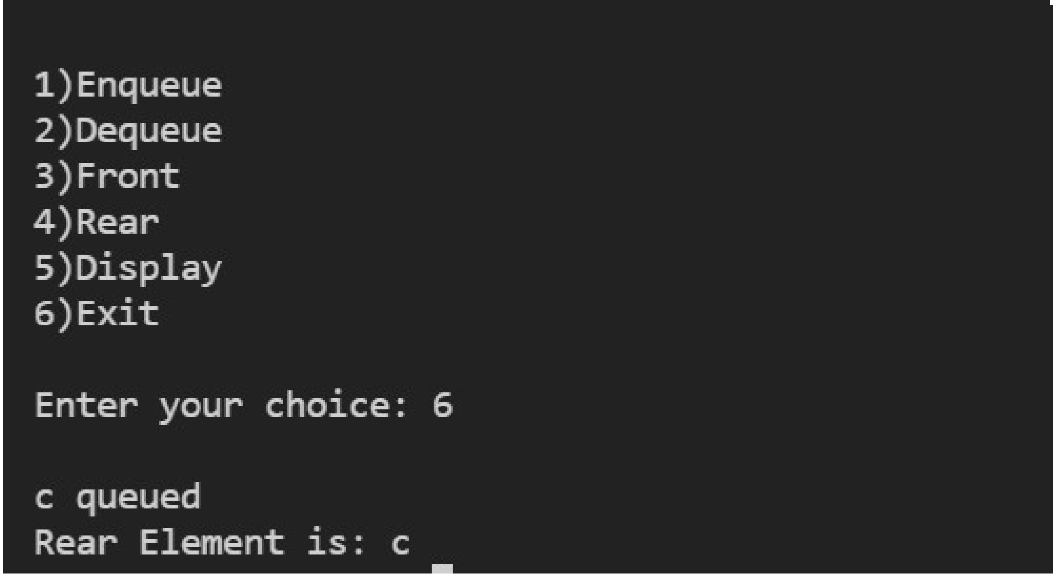
OUTPUT



```
99 queued
Front Element is: 99
Rear Element is: 99


1)Enqueue
2)Dequeue
3)Front
4)Rear
5)Display
6)Exit

Enter your choice: 1
```



```
Enter that you to enqueue in the stack: 34.32

34.320000 queued


1)Enqueue
2)Dequeue
3)Front
4)Rear
5)Display
6)Exit

Enter your choice: 5

The whole queue is

           |
           |rear
           |
           v
|             34.3200|
|                  99|
          |
          |front
          |
          v
```



```
1)Enqueue
2)Dequeue
3)Front
4)Rear
5)Display
6)Exit

Enter your choice: 6

c queued
Rear Element is: c
```

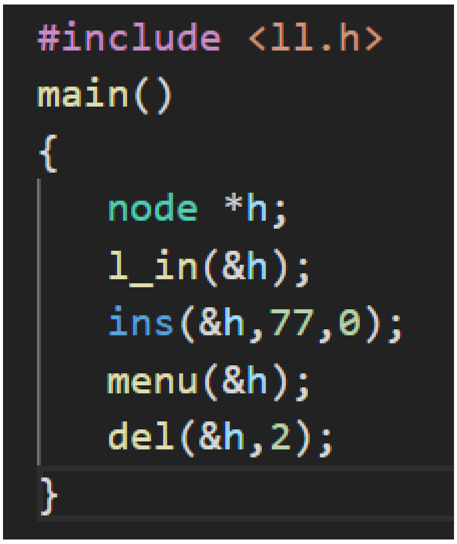
CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE

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

main.c

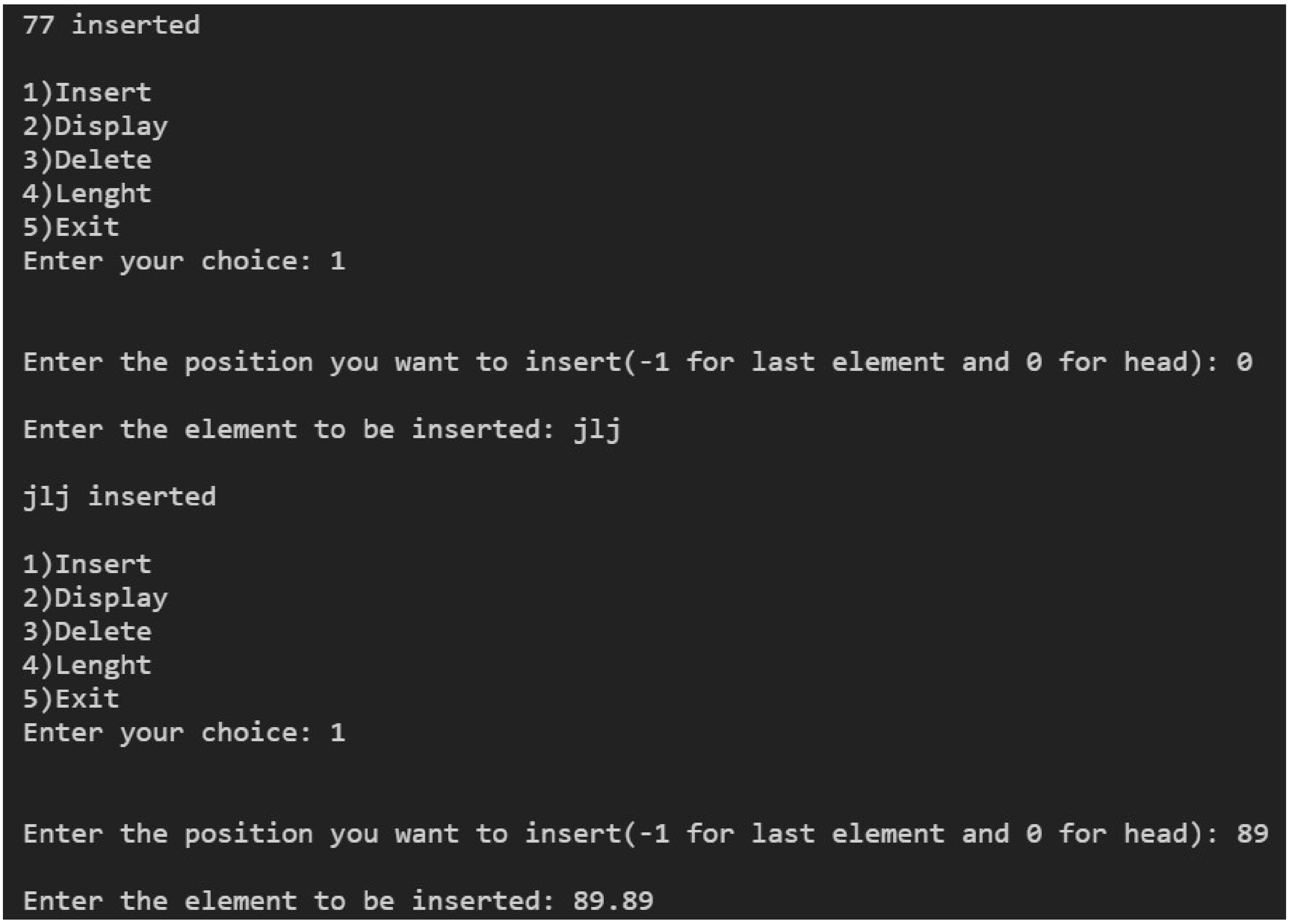
CODE

```
#include <ll.h>

main()

{

    node *h;

    l_in(&h);

    ins(&h,77,0);

    menu(&h);

    del(&h,2);

}
```

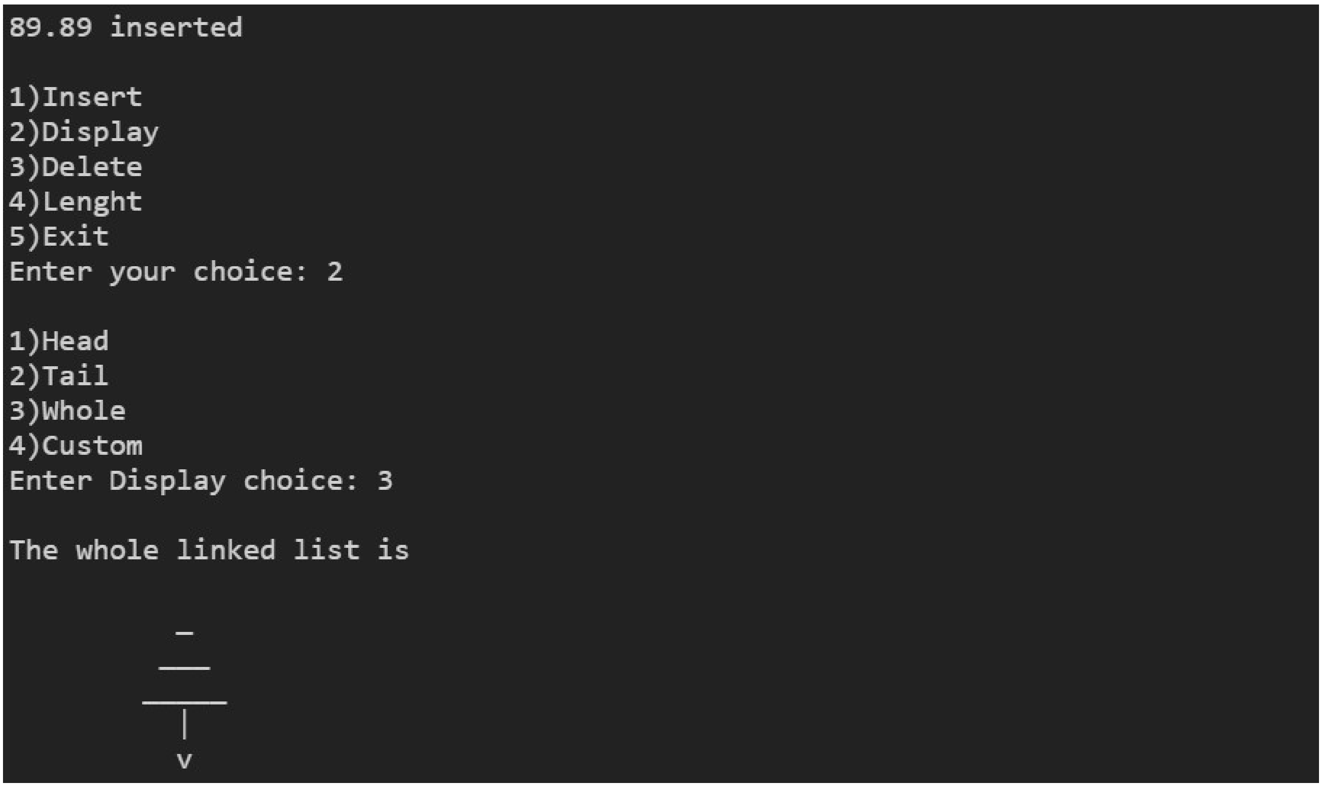


```
#include <ll.h>
main()
{
    node *h;
    l_in(&h);
    ins(&h,77,0);
    menu(&h);
    del(&h,2);
}
```

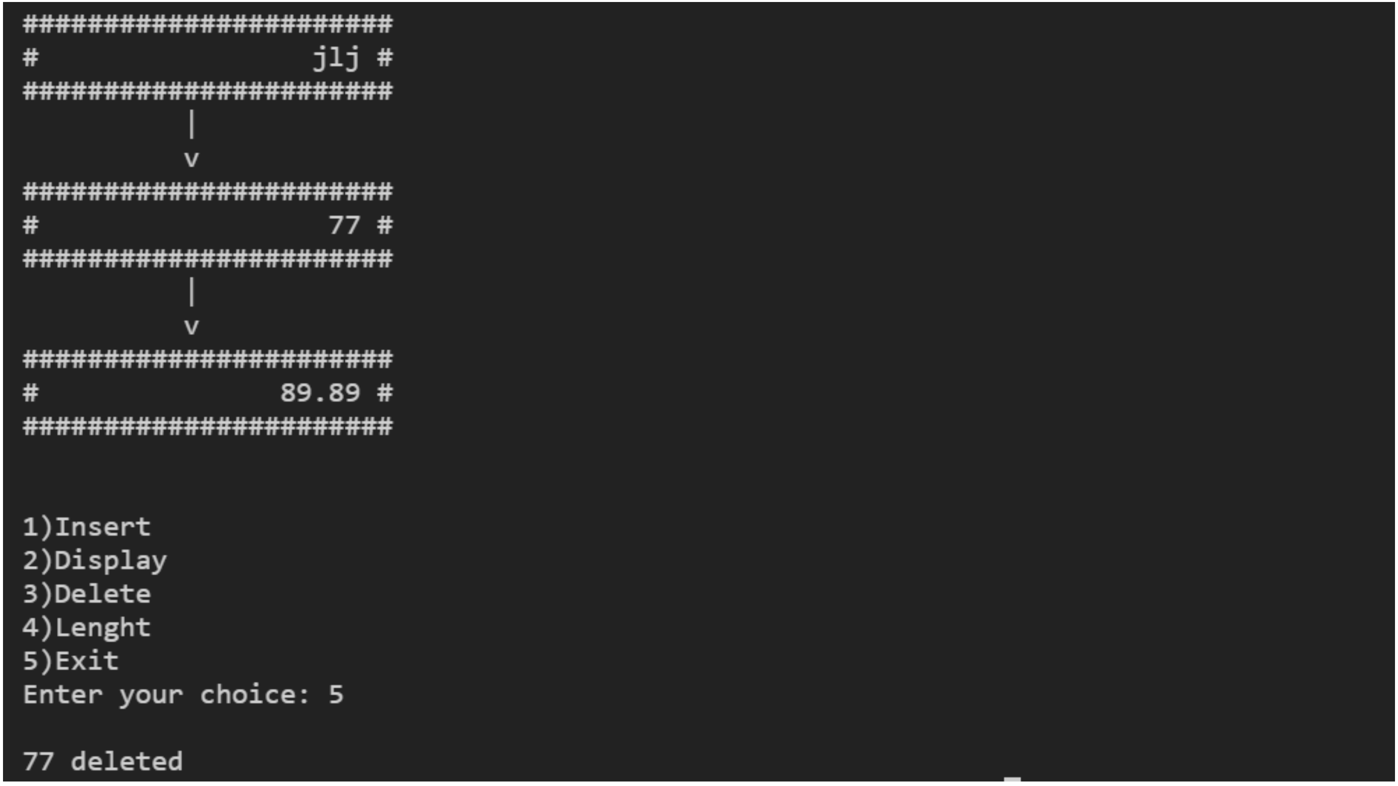
OUTPUT



```
77 inserted

1)Insert
2)Display
3)Delete
4)Lenght
5)Exit
Enter your choice: 1


Enter the position you want to insert(-1 for last element and 0 for head): 0

Enter the element to be inserted: jlj

jlj inserted

1)Insert
2)Display
3)Delete
4)Lenght
5)Exit
Enter your choice: 1


Enter the position you want to insert(-1 for last element and 0 for head): 89

Enter the element to be inserted: 89.89
```



```
89.89 inserted

1)Insert
2)Display
3)Delete
4)Lenght
5)Exit
Enter your choice: 2

1)Head
2)Tail
3)Whole
4)Custom
Enter Display choice: 3

The whole linked list is


          _
         ___
        _____
          |
          v
```



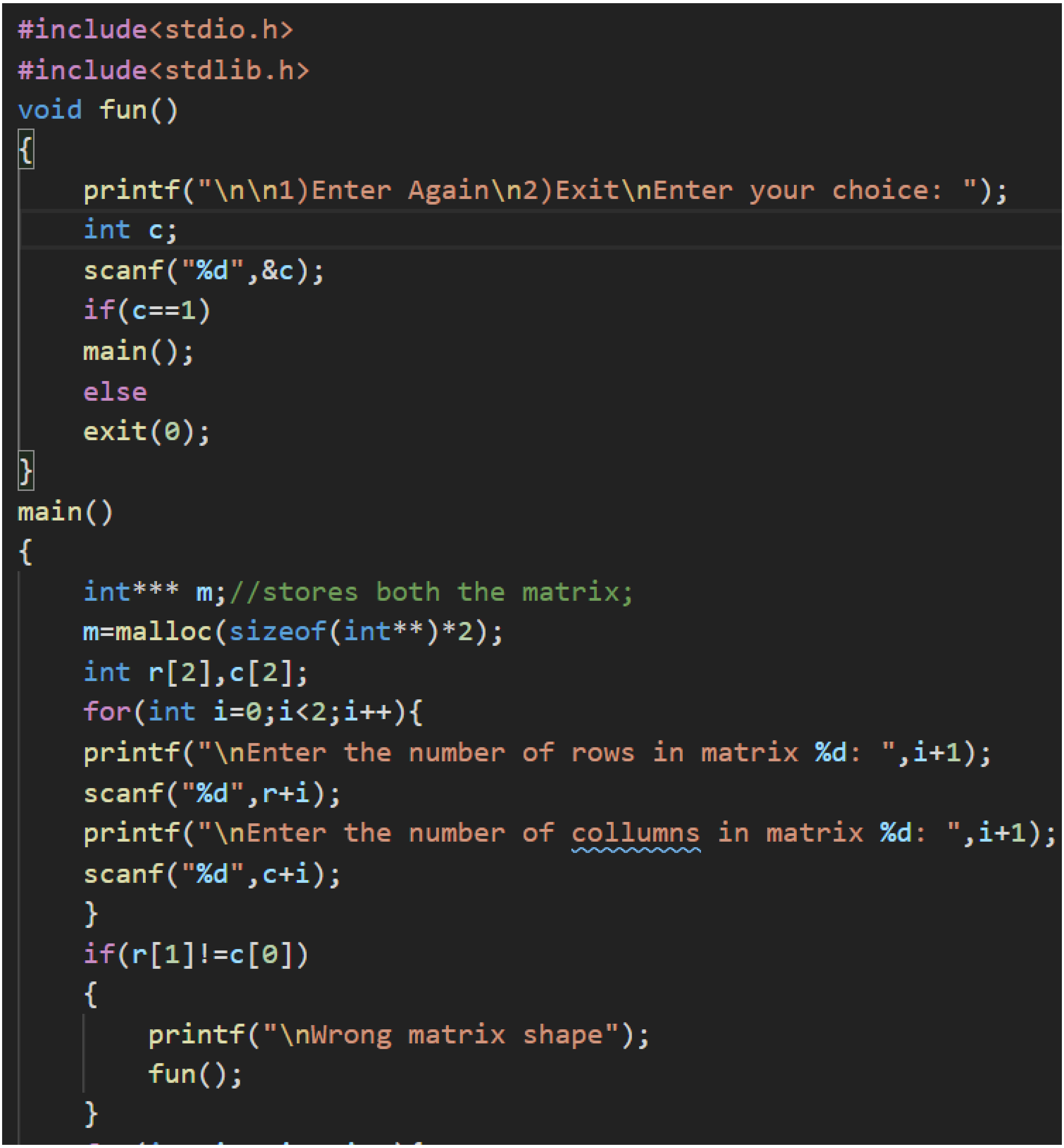
```
#######################
#                 jlj #
#######################
          |
          v
#######################
#                  77 #
#######################
          |
          v
#######################
#               89.89 #
#######################


1)Insert
2)Display
3)Delete
4)Lenght
5)Exit
Enter your choice: 5

77 deleted
```

CLICK HERE FOR GITHUB LINK OF WHOLE SOURCE CODE

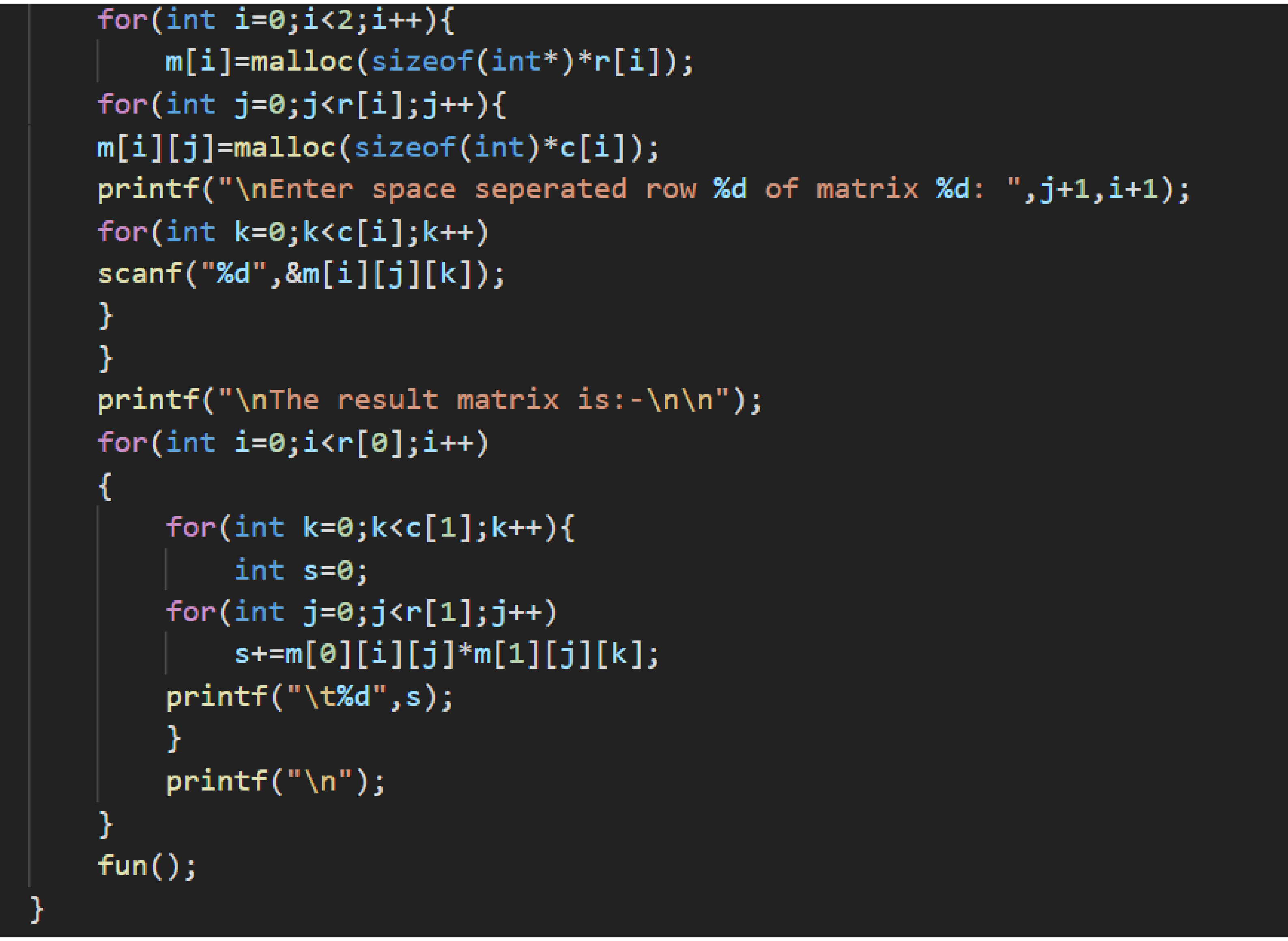
Q4) Multiply two matrices of order (m X n) and (n X p). If the order of the matrices is other than this, then supply an error message.

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();
}
```

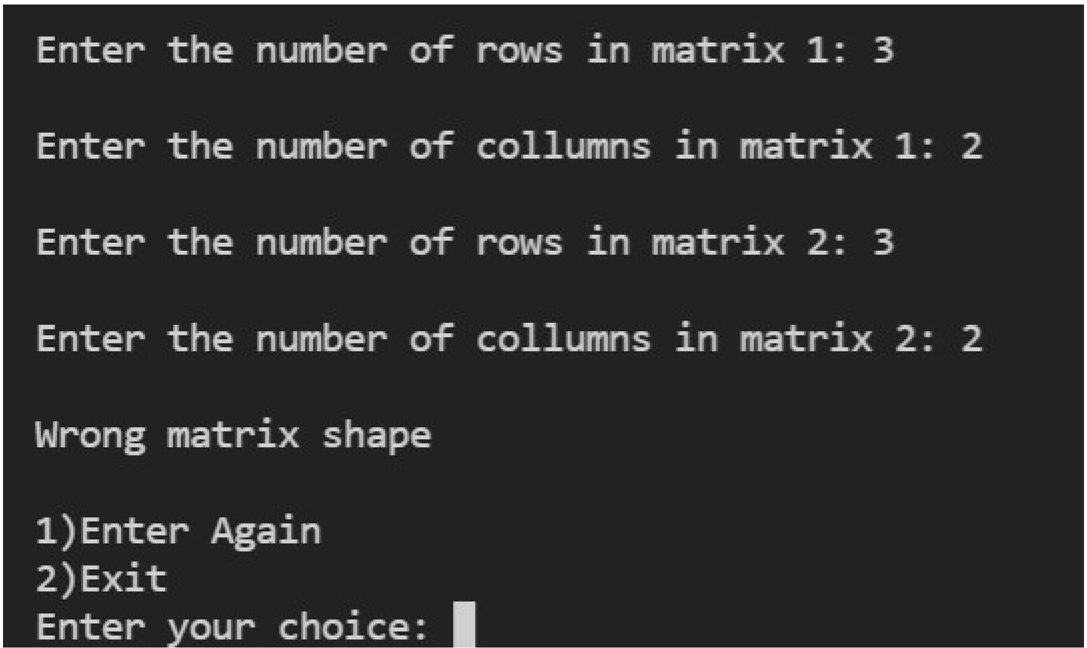


```c
#include<stdio.h>
#include<stdlib.h>
void fun()
{
    printf("\n\n1)Enter Again\n2)Exit\nEnter your choice: ");
    int c;
    scanf("%d",&c);
    if(c==1)
    main();
    else
    exit(0);
}
main()
{
    int*** m;//stores both the matrix;
    m=malloc(sizeof(int**)*2);
    int r[2],c[2];
    for(int 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();
    }
```

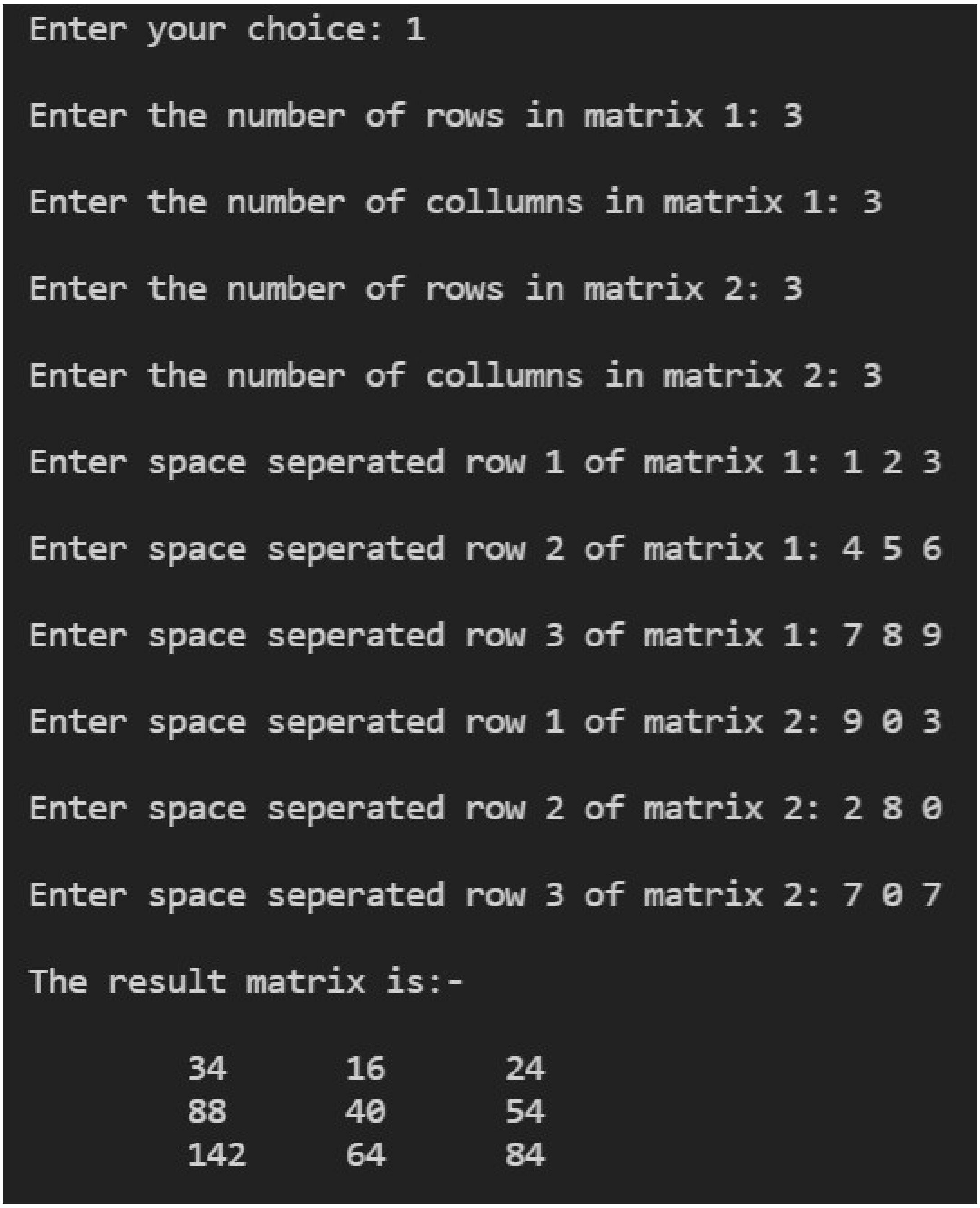


```c
    for(int i=0;i<2;i++){
        m[i]=malloc(sizeof(int*)*r[i]);
    for(int j=0;j<r[i];j++){
    m[i][j]=malloc(sizeof(int)*c[i]);
    printf("\nEnter space seperated row %d of matrix %d: ",j+1,i+1);
    for(int k=0;k<c[i];k++)
    scanf("%d",&m[i][j][k]);
    }
    }
    printf("\nThe result 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

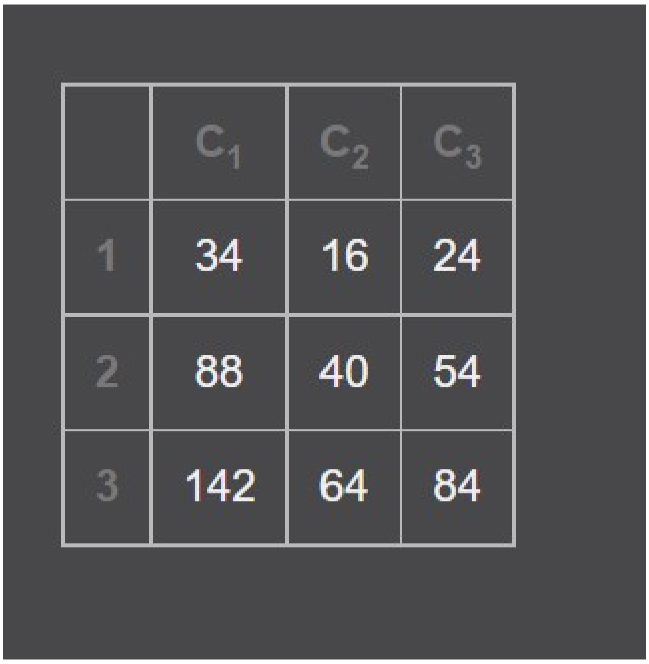


```
Enter the number of rows in matrix 1: 3

Enter the number of collumns in matrix 1: 2

Enter the number of rows in matrix 2: 3

Enter the number of collumns in matrix 2: 2

Wrong matrix shape

1)Enter Again
2)Exit
Enter your choice:
```



```
Enter your choice: 1

Enter the number of rows in matrix 1: 3

Enter the number of collumns in matrix 1: 3

Enter the number of rows in matrix 2: 3

Enter the number of collumns in matrix 2: 3

Enter space seperated row 1 of matrix 1: 1 2 3

Enter space seperated row 2 of matrix 1: 4 5 6

Enter space seperated row 3 of matrix 1: 7 8 9

Enter space seperated row 1 of matrix 2: 9 0 3

Enter space seperated row 2 of matrix 2: 2 8 0

Enter space seperated row 3 of matrix 2: 7 0 7

The result matrix is:-

        34      16      24
        88      40      54
        142     64      84
```

CALCULATED USING CALCULATOR



	C_1	C_2	C_3
1	34	16	24
2	88	40	54
3	142	64	84

CLICK HERE FOR GITHUB LINK

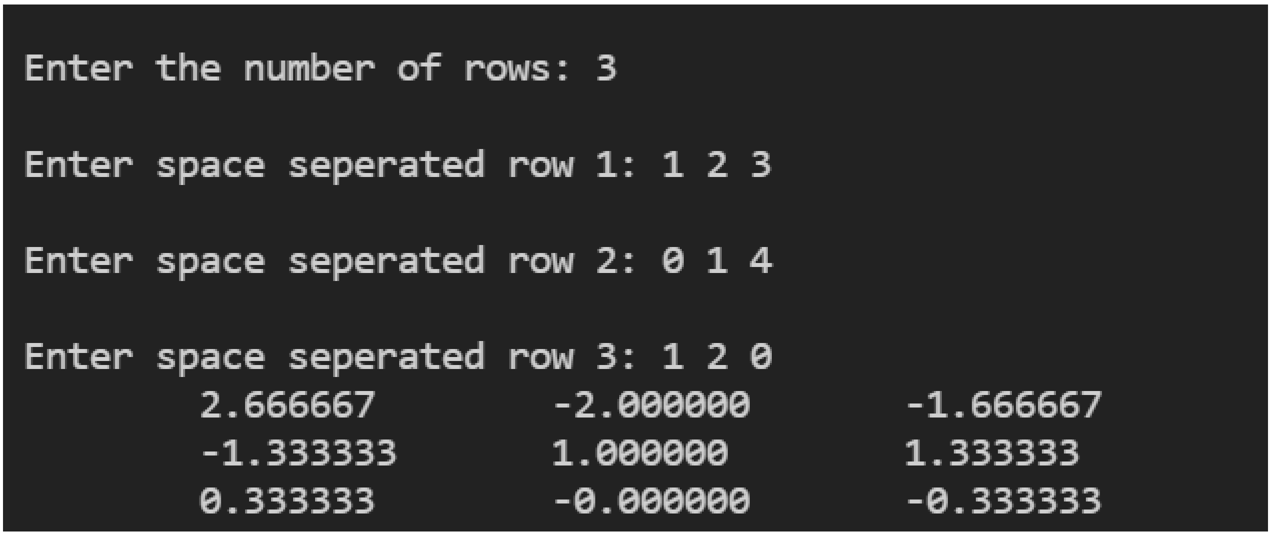
Q5) Compute the inverse of a square matrix of order (n X n). If the matrix is a singular matrix, then supply appropriate error message.

main.c

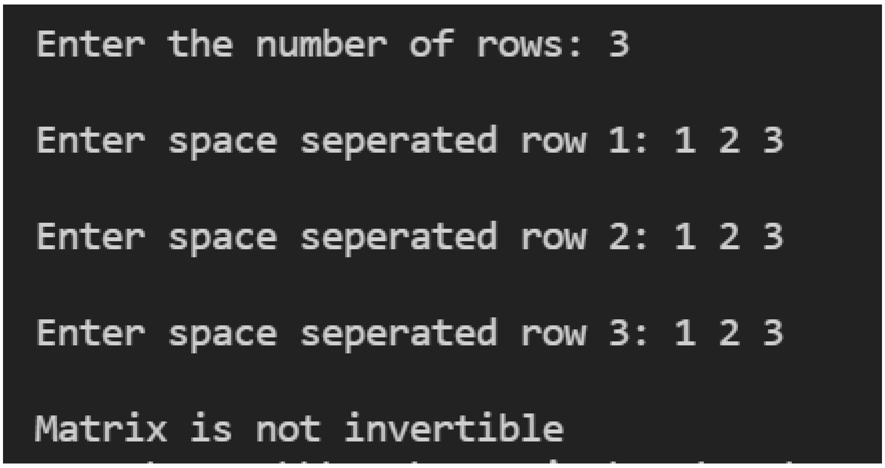
CODE

```
#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



```
Enter the number of rows: 3

Enter space seperated row 1: 1 2 3

Enter space seperated row 2: 0 1 4

Enter space seperated row 3: 1 2 0
        2.666667        -2.000000       -1.666667
        -1.333333       1.000000        1.333333
        0.333333        -0.000000       -0.333333
```



```
Enter the number of rows: 3

Enter space seperated row 1: 1 2 3

Enter space seperated row 2: 1 2 3

Enter space seperated row 3: 1 2 3

Matrix is not invertible
```

CLICK HERE FOR GITHUB LINK