ST. XAVIER’S COLLEGE **Maitighar, Kathmandu**

****

**Data Structure and Algorithms**

**Lab Assignment # 10**

**Submitted by:**

**Swagat Niroula**

**B.Sc.CSIT 2nd Semester**

**014BSCIT048**

**Submitted to:**

**Balkrishna Subedi**

**Lecturer**

**Department of Computer Science**

**St. Xavier’s College**

**Assignment # 10.1**

**STATEMENT**

WAP TO IMPLEMENT LINEAR QUEUE OPERATION

**ALGORITHM**

**SOURCE CODE**

#include<stdio.h>

#include<conio.h>

#include<process.h>

#define MAX 50

int q\_item[MAX];

int r = - 1;

int f = - 1;

void ins(void);

void del(void);

void dis(void);

void main()

{

int c;

up:

clrscr();

printf("Queue Operations\n");

printf("----------------\n\n");

printf("1.Insert element\n");

printf("2.Delete element\n");

printf("3.Display all elements\n");

printf("4.Quit \n");

printf("\n\tPlease enter your choice: ");

scanf("%d", &c);

switch (c)

{

case 1:

{

ins();

break;

}

case 2:

{

del();

break;

}

case 3:

{

dis();

break;

}

case 4:

{

exit(1);

}

default:

{

printf("Wrong choice\n");

getch();

}

}

goto up;

}

void ins()

{

int add\_item;

if (r == MAX - 1)

{

printf("Queue Overflow \n");

getch();

}

else

{

if (f == - 1)

f = 0;

printf("Inset the element in queue : ");

scanf("%d", &add\_item);

r = r + 1;

q\_item[r] = add\_item;

}

}

void del()

{

if (f == - 1 || f > r)

{

printf("Queue Underflow \n");

getch();

return ;

}

else

{

printf("Element deleted from queue is : %d\n", q\_item[f]);

f = f + 1;

getch();

}

}

void dis()

{

int i;

if (f == - 1)

{

printf("Queue is empty \n");

getch();

}

else

{

printf("Queue is : \n");

for (i = f; i <= r; i++)

{

printf("%d ", q\_item[i]);

printf("\n");

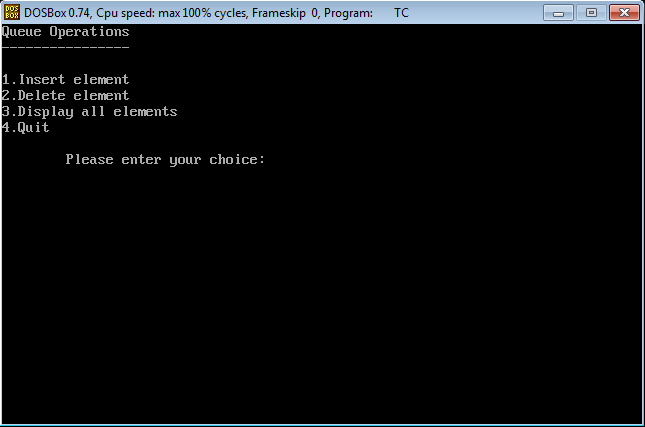
}

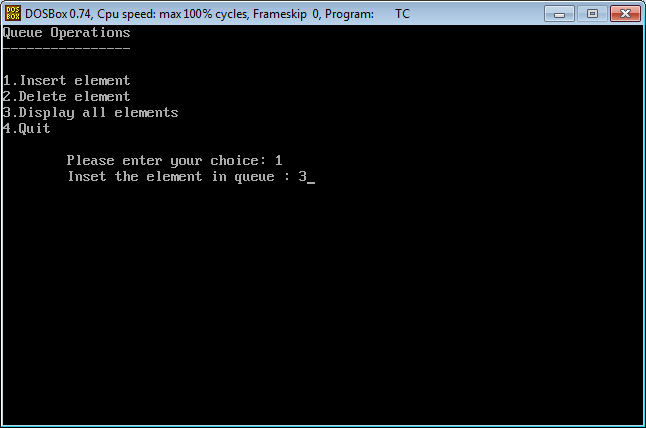
getch();

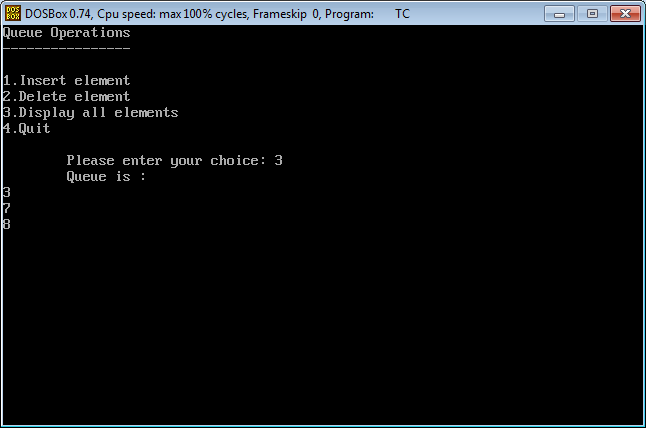
}

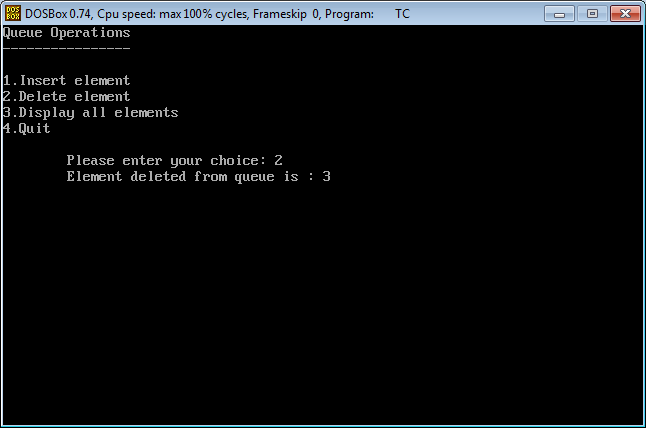
}

**OUTPUT**

****

****

****

****

**CONCLUSION**

Hence, the implementation of linear queue can be done, using array and function.

**Assignment # 9.2**

**STATEMENT**

WAP TO FIND THE SUM OF NATURAL NUMBER USING RECURSION

**ALGORITHM**

Main Function

Step 1: Start the Program

Step 2: Input an integer

Step 3: Call sum() function with argument x

Step 4: End the Program

sum Function

Step 1: Start the Function

Step 2: If argument x is 1 return 1

Else return the sum of x and return value of fact function with argument x -1

Step 3: End the Function

**SOURCE CODE**

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int x;

long int sum(int);

printf("\nPlease input any number: ");

scanf("%d",&x);

printf("\nFactorial of %d is %d",x,sum (x));

getch();

}

long int sum(int x)

{

if(x==1)

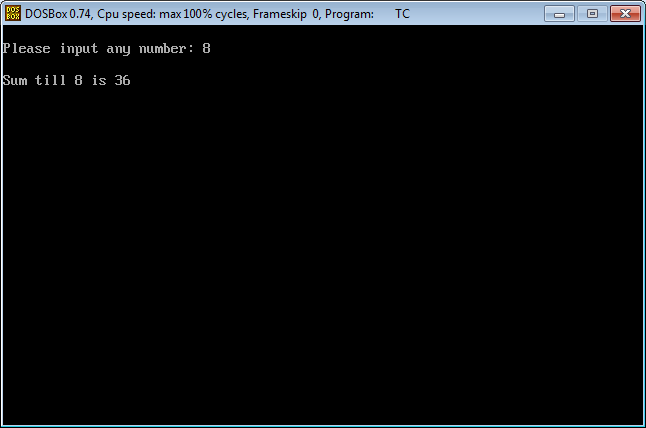
return 1;

else

return(x+fact(x-1));

}

**OUTPUT**

****

**CONCLUSION**

Hence, the sum of natural number can be found using recursion.

**Assignment # 9.3**

**STATEMENT**

WAP TO EVALUATE VALUE OF GIVEN EXPRESSION

+ A \* B C

**ALGORITHM**

**SOURCE CODE**

**OUTPUT**

**CONCLUSION**

Hence, the value of the given expression can be evaluated using stack operation.