

Oxford College of Engineering and Management

Lab Report

Data Structure and Algorithm

Course Code: CMP 227

BCA Second Year, Third Semester

Submitted To:

Asst. Professor Suresh Raj Sharma

Department Of BCA

Submitted By:

Debid Rana Magar

Section: A

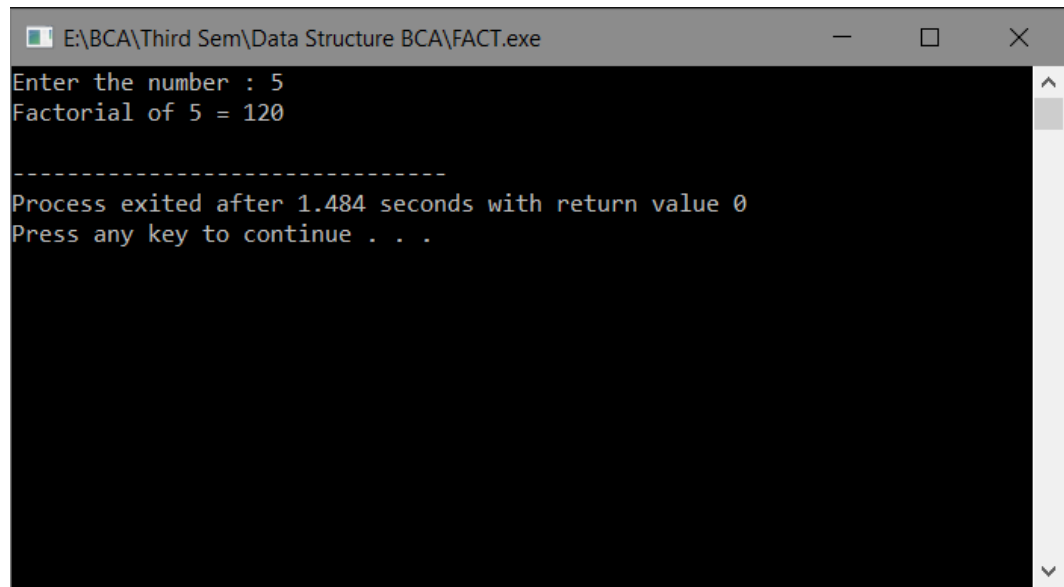
Roll No: 25

Lab 1

Factorial of given number using recursion

```
#include<stdio.h>
#include<conio.h>
int factorial(int);
main()
{
    int n,value;
    printf( "Enter the number : ");
    scanf( "%d", &n );
    if ( n < 0 )
        printf( "No factorial of negative number\n");
    else
        if ( n==0 )
            printf( "Factorial of zero is 1\n" );
        else
        {
            value = factorial(n);
            printf( "Factorial of %d = %d\n", n, value );
        }
}
int factorial(int k )
{
    int fact =1;
    if ( k > 1 )
        fact = k * factorial( k-1 ); /* Recursive function call */
    return ( fact );
}
```

Output:



```
E:\BCA\Third Sem\Data Structure BCA\FACT.exe
Enter the number : 5
Factorial of 5 = 120

-----
Process exited after 1.484 seconds with return value 0
Press any key to continue . . .
```

Lab 2

Program of stack to show PUSH and POP operation

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#define MAX 5
int push();
int pop();
int display();
int top = -1;
int stack_arr[MAX];
```

```

main()
{
    system("cls");
    int choice;
    while(1)
    {
        printf("\n1.Push\n");
        printf("2.Pop\n");
        printf("3.Display\n");
        printf("4.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                exit(1);
            default:
                printf("Wrong choice\n");
        }/*End of switch*/
    }/*End of while*/
}/*End of main()*/

```

```

push()
{
    int pushed_item;
    if(top == (MAX-1))
        printf("Stack Overflow\n");
    else
    {
        printf("Enter the item to be pushed in stack : ");
        scanf("%d",&pushed_item);
        top=top+1;
        stack_arr[top] = pushed_item;
        display();
    }
    return 0;
}/*End of push()*/

```

```

pop()
{
    if(top == -1)
        printf("Stack Underflow\n");
    else
    {
        printf("Popped element is : %d\n",stack_arr[top]);
        top=top-1;
    }
    return 0;
}/*End of pop()*/

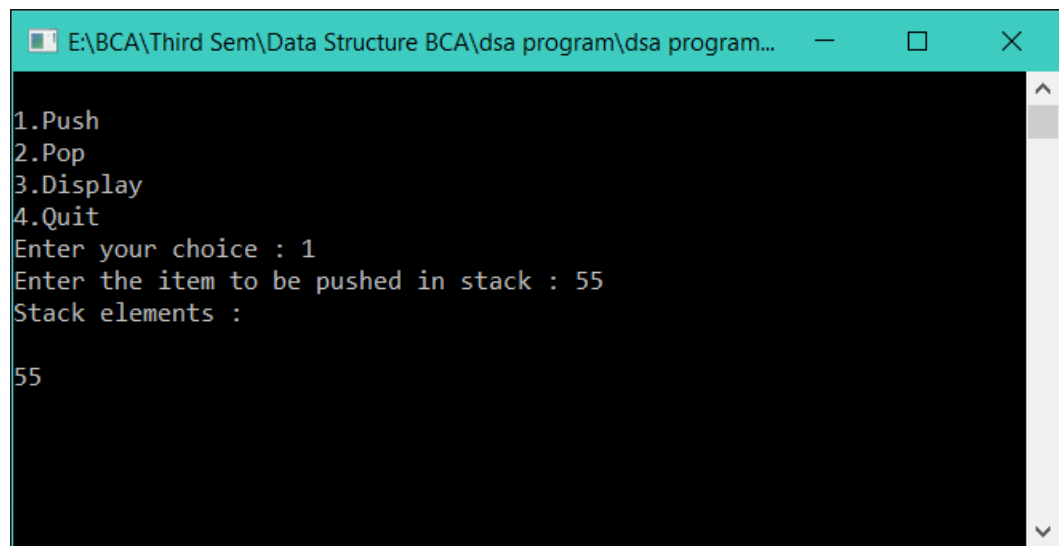
```

```

display()
{
    int i;
    if(top == -1)
        printf("Stack is empty\n");
    else
    {
        printf("Stack elements :\n");
        for(i = top; i >=0; i--)
            printf("\n%d\t", stack_arr[i] );
    }
    getch();
}/*End of display()*/

```

Output:



```

E:\BCA\Third Sem\Data Structure BCA\dsa program\dsa program...
1.Push
2.Pop
3.Display
4.Quit
Enter your choice : 1
Enter the item to be pushed in stack : 55
Stack elements :
55

```

Lab 3

Tower of Hanoi

```
/* Program for solution of Tower of Hanoi*/
#include<stdio.h>
int toh(int,char,char,char);
int main()
{
    char source= 'S',temp= 'T ', dest= 'D';
    int ndisk;
    printf("Enter the number of disks : ");
    scanf ( "%d", &ndisk );
    printf ("Sequence is :\n");
    toh( ndisk, source, temp, dest );
    return 0;
}
toh( int ndisk, char source, char temp, char dest )
{
    if ( ndisk > 0 )
    {
        toh ( ndisk-1, source, dest, temp );
        printf ( "Move Disk %d  %c-->%c\n", ndisk, source,dest );
        toh( ndisk-1, temp, source, dest );
    }
    return 0;
}/*End of toh()*/
```

Output:

```
E:\BCA\Third Sem\Data Structure BCA\T_HANOI.exe
Enter the number of disks : 3
Sequence is :
Move Disk 1 S-->D
Move Disk 2 S-->
Move Disk 1 D-->
Move Disk 3 S-->D
Move Disk 1 -->S
Move Disk 2 -->D
Move Disk 1 S-->D
-----
Process exited after 2.883 seconds with return value 0
Press any key to continue . . .
```

Lab 4

Program of single linked list

```
#include<stdio.h>
#include<process.h>
#include<malloc.h>
#include<stdlib.h>
int create_list(int);
int addatbeg(int);
int addafter(int,int);
int del(int);
int display();

struct node
{
    int info;
    struct node *link;
}*start;
```



```

main()
{
    int choice,n,m,position,i;
    start=NULL;
    while(1)
    {
        printf("1.Create List\n");
        printf("2.Add at begining\n");
        printf("3.Add after \n");
        printf("4.Delete\n");
        printf("5.Display\n");
        printf("6.Quit\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                printf("How many nodes you want : ");
                scanf("%d",&n);
                for(i=0;i<n;i++)
                {
                    printf("Enter the element : ");
                    scanf("%d",&m);
                    create_list(m);
                }
                break;
            case 2:
                printf("Enter the element : ");
                scanf("%d",&m);
                addatbeg(m);

```

```

        break;
    case 3:
        printf("Enter the element : ");
        scanf("%d",&m);
        printf("Enter the position after which this element
is inserted : ");
        scanf("%d",&position);
        addafter(m,position);
        break;
    case 4:
        if(start==NULL)
        {
            printf("List is empty\n");
            continue;
        }
        printf("Enter the element for deletion : ");
        scanf("%d",&m);
        del(m);
        break;
    case 5:
        display();
        break;
    case 6:
        exit(0);
    default:
        printf("Wrong choice\n");
}/*End of switch */
}/*End of while */
}/*End of main()*/

```

```

int create_list(int data)

```

```

{
    struct node *q,*tmp;
    tmp= malloc(sizeof(struct node));
    tmp->info=data;
    tmp->link=NULL;

    if(start==NULL) /*If list is empty */
        start=tmp;
    else
    { /*Element inserted at the end */
        q=start;
        while(q->link!=NULL)
            q=q->link;
        q->link=tmp;
    }
}/*End of create_list()*/

```

```

addatbeg(int data)
{
    struct node *tmp;
    tmp=malloc(sizeof(struct node));
    tmp->info=data;
    tmp->link=start;
    start=tmp;
}/*End of addatbeg()*/

```

```

addafter(int data,int pos)
{
    struct node *tmp,*q;
    int i;

```

```

q=start;
for(i=0;i<pos-1;i++)
{
    q=q->link;
    if(q==NULL)
    {
        printf("There are less than %d elements",pos);
        return;
    }
}/*End of for*/

tmp=malloc(sizeof(struct node) );
tmp->link=q->link;
tmp->info=data;
q->link=tmp;
}/*End of addafter()*/

del(int data)
{
    struct node *tmp,*q;
    if(start->info == data)
    {
        tmp=start;
        start=start->link; /*First element deleted*/
        free(tmp);
        return;
    }
    q=start;
    while(q->link->link != NULL)
    {

```

```

        if(q->link->info==data)    /*Element deleted in between*/
        {
            tmp=q->link;
            q->link=tmp->link;
            free(tmp);
            return;
        }
        q=q->link;
    }/*End of while */
    if(q->link->info==data)    /*Last element deleted*/
    {
        tmp=q->link;
        free(tmp);
        q->link=NULL;
        return;
    }
    printf("Element %d not found\n",data);
}/*End of del()*/

```

```

display()
{
    struct node *q;
    if(start == NULL)
    {
        printf("List is empty\n");
        return;
    }
    q=start;
    printf("List is :\n");
    while(q!=NULL)

```

```

        {
            printf("%d ", q->info);
            q=q->link;
        }
        printf("\n");
    }/*End of display() */

```

Output:

```

3.Add after
4.Delete
5.Display
6.Quit
Enter your choice : 1
How many nodes you want : 3
Enter the element : 55
Enter the element : 88
Enter the element : 33
1.Create List
2.Add at begining
3.Add after
4.Delete
5.Display
6.Quit
Enter your choice : 5
List is :
55 88 33

```

Lab 5

Fibonacci Series Using Recursion

```
#include <stdio.h>
#include<conio.h>
int fibo(int);

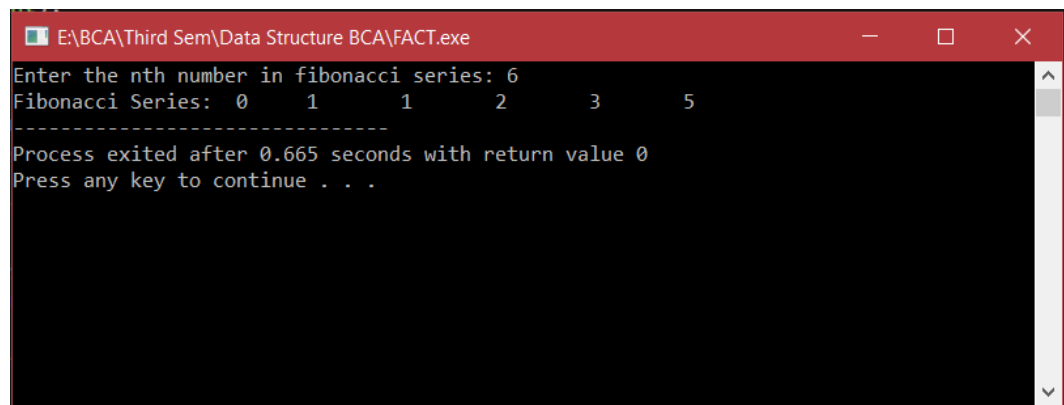
int main()
{
    int num;
    int result, i;

    printf("Enter the nth number in fibonacci series: ");
    scanf("%d", &num);
    printf("Fibonacci Series: ");
    if (num < 0)
    {
        printf("Fibonacci of negative number is not possible.\n");
    }
    else
    {
        for(i=0;i<num;i++)
        {
            result = fibo(i);
            printf(" %d \t", result);
        }
    }
    return 0;
}

int fibo(int num)
```

```
{
    if (num == 0)
    {
        return 0;
    }
    else if (num == 1)
    {
        return 1;
    }
    else
    {
        return(fibo(num - 1) + fibo(num - 2));
    }
}
```

Output:



The screenshot shows a Windows command prompt window with a red title bar. The title bar text is "E:\BCA\Third Sem\Data Structure BCA\FACT.exe". The window contains the following text: "Enter the nth number in fibonacci series: 6", "Fibonacci Series: 0 1 1 2 3 5", a dashed line separator, "Process exited after 0.665 seconds with return value 0", and "Press any key to continue . . .". The text is displayed in a monospaced font on a black background.