MASTER OF COMPUTER APPLICATIONS

PRACTICAL RECORD WORK

ON

20MCA135 DATA STRUCTURES LAB

Submitted

By

…….

(Reg No : …….)

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING VADAKARA

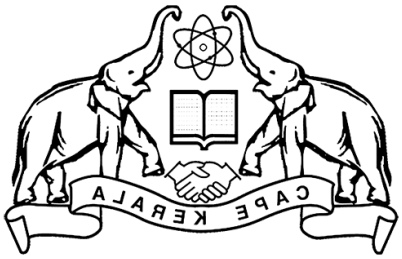
(CAPE – GOVT.OF KERALA)

DECEMBER 2021

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING VADAKARA

(CAPE – GOVT.OF KERALA)



CERTIFICATE

Certified that this is a bonafide record of the practical work on the course 20MCA135 DATA STRUCTURE LAB done by Mr/Mrs……. (Reg No:……..) First semester MCA student of Department of Computer Applications at College of Engineering Vadakara in the practical fulfillment for the award of the degree of Master of Computer Applications (MCA) of APJ Abdul Kalam Technological University (KTU)

(Mrs. Bindu S Mony)

FACULTY-IN-CHARGE **HEAD OF THE DEPARTMENT**

DATE:

EXAMINERS:

INDEX			
SL.NO	**PROGRAMS**	**PAGE NO**	**REMARKS**
1	Implementation of Array	4	
2	Array deletion	5	
3	Exchange sort	7	
4	Merge 2 arrays	8	
5	Linear search	10	
6	Binary search	12	
7	Stack using array	14	
8	Queue using array	17	
9	Circular queue	21	
10	Queue using stack	28	
11	Sum of elements in odd position	33	
12	Sum and difference of matrix	34	
13	Transpose of matrix	38	
14	Fibonacci up to a range	40	
15	Matrix multiplication	41	

1. Implementation of Array

```
#include<stdio.h>
Void main()
{
  Int a[10],I,n,os,num;
  Printf(“Enter the No. of elements:”);
  Scanf(“%d”,&n);
  Printf(“Enter the elements:”);
  For(i=0;i<n;i++)
  {
    Scanf(“%d”,&a[i]);
  }
  n++;
  printf(“Enter the Position”);
  scanf(“%d”,&pos);
  printf(“Enter the No. to be inserted:”);
  scanf(“%d”,num);
  for(i=n-1;i>=pos;i--)
  {
      a[i]=a[i-1];
      a[pos-1]=num;
  }
  Printf(“New Array:”);
  For(i=0;i<n;i++)
  {
```

```
    printf(“%d”,&a[i]);
  }
}
```

OUTPUT:

```
Enter the No. of element: 5
Enter the Elements:
 1 2 3 4 5
Enter the Position: 3
Enter the NO to be inserted: 7
1 2 7 3 4 5
```

2. Implementation of Array deletion

```
#include<stdio.h>
 void main()
{
    int a[10],i,n,pos;
    printf("Enter no of elements \n");
    scanf("%d",&n);
    printf("Enter the elements \n");
    for (i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
```

```
    }
    printf("Elements of array are\n");
    for(i=0;i<n;i++)
    {
       printf("a[%d] = %d\n",i,a[i]);
    }

    printf("Enter the position of number has to be deleted\n");
    scanf("%d",&pos);
    for(i=pos;i<n-1;i++)
    {
       a[i]=a[i+1];
    }
    n=n-1;
    printf("Array after deletion:");
    for(i=0;i<n;i++)
    {
       printf("%d”,a[i]);
    }
}
```

OUTPUT:

Enter the No of Elements: 4

Enter the element:

1 2 3 4

Enter the postion of No. to be delete: 2

Array after deletion:

1 3 4

3. Implementation of Exchange sort

```
#linclude<stdio.h>
void  main()
{
   int i,j,n,a[10],temp;
   printf("Enter the number of elements:\n") ;
   scanf("%d",&n) ;
   printf("Enter the elements\n") ;
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]) ;
   }
   for(i=0;i<n;i++)
    {
       for(j=i+1;j<n;j++)
       {
           if(a[i]>a[j])
           {
              temp=a[i];
               a[i]=a[j];
               a[j]=temp;
```

```
            }
        }
    }

  printf("Elements sorted in ascending order are\n");
    for(i=0;i<n;i++)
    {
        printf("%d ",a[i]) ;
    }
}
```

OUTPUT:

```
Enter the No. of elements: 5
Enter the elements:
4   21   36   14   5
Elements sorted in ascending order are:
4   5   14   21   36
```

4.Merge two arrays

```
#include<stdio.h>
Void main()
{
```

```
int a[50], b2[50], n,m,i, p;
printf("Enter Array 1 Size: ");
scanf("%d", &n);
printf("Enter Array 1 Elements: ");
for(i=0; i<n; i++)
{
   scanf("%d", &a[i]);
 }
printf("\nEnter Array 2 Size: ");
scanf("%d", &m);
printf("Enter Array 2 Elements: ");
for(i=0; i<m; i++)
{
   scanf("%d", &b[i]);
}
P=n+m;
for(i=0; i<n; i++)
{
    C[i]=a[i];
 }
 for(i=n; i<p; i++)
{
    C[i]=b[i];
 }
 printf("The new array after merging is:\n");
 for(i=0; i<n; i++)
```

```
    {
        scanf("%d", &a[i]);
     }
}
```

OUTPUT:

```
Enter Array 1 Size: 3
Enter Array 1 Elements:
1 2 3
Enter Array 2 Size:3
Enter Array 2 Elements:4 5 6
The new array after merging:
1 2 3 4 5 6
```

5.Searching an element using Linear Search

```
#include<stdio.h>
void  main()
{
   int i,n,a[100],search,flag=0;
   printf("Enter the number of elements:") ;
   scanf("%d",&n) ;
   printf("Enter the elements\n") ;
   for(i=0;i<n;i++)
   {
```

```
    scanf("%d",&a[i]) ;
  }

  printf("Enter the element to be searched\n");
  scanf("%d",&search);

  for(i=0;i<n;i++)
  {
    if(a[i]==search)
    {
      printf("Element %d found at %d position\n",search,i);
      flag=1;
      break;
    }
  }
  if(flag==0)
  {
    printf("Element %d not found\n",search);
  }
}
```

OUTPUT:

```
Enter the no of elements:4
Enter the elements:
```

1 7 4 3

Enter the No to be searched: 4

Element 4 found at 3 position

6. Searching an element using Binary Search

```
#include<stdio.h>
void main()
{
   int a[10],i,n,item,flag=0,f,mid;
   printf("\n  Enter the size of an array: ");
   scanf("%d",&n);
   printf("\n  Enter the elements in ascending order: ");
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }

   printf("\n  Enter the number to be search: ");
   scanf("%d",&item);

   f=0,l =n-1;
   for(i=0i<ni++)
   {
       Mid=(f+l)/2;
```

```
        If(a[m]==item)
         {
             Flag=1;
              t==m+1;
         }
     elseif(a[m]<s)
     {
          f=m+1;
      }
     else
     {
           m=l-1;
      }
   }
    if(flag==1)
    printf("The number is found and its position is: %d",t");
    else
    {
    printf("Element not found  );
    }
}
```

OUTPUT:

Enter the no of elements:4

Enter the elements:

1 7 4 3

Enter the No to be searched: 4

Element 4 found at 3 position

7. Implementation of Stack using array

```
#include<stdio.h>
int SIZE = 3;
void push(int a[],int *last)
{
 if(*last >= SIZE - 1)
 {
 printf("stack overflow");
 }
 else
 {
*last +=1;
printf("\n Enter the element");
scanf("%d",&a[*last]);

printf("\n%d pushed in to the stack",a[*last]);  }

}
int pop(int a[],int *last)
{
int ele;
 if(*last > -1)
 {
ele = a[*last];
*last -=1;
printf("\n The value %d poped from the stack", ele);  }
 else
 printf("stack underflow");
}
```

```
void display(int a[], int *last)
{
int i;

if(*last < 0)
{
 printf("\n stack is empty");
 return;
}
else
{
printf("\n The stack elements are:");
for(i=0;i <= *last;i++)
        printf("%d ",a[i]);
}
}
int main()
{
int arr[SIZE],ch,e=1;
int last=-1;

while(e)
{
printf("\n STACK USING ARRAY");
printf("\n.....MENU.....");
printf("\n 1.Push \n 2.Pop \n 3.Display \n 4.Exit");
printf("\n.....");
printf("\n Enter your choice");
scanf("%d",&ch);
switch(ch)
{
        case 1:
                push(arr,&last);
                break;
        case 2:
                pop(arr,&last);
                break;
        case 3:
                display(arr,&last);
```

```
            break;
        case 4:
            e=0;
            printf("\n exiting");
            break;
        default: printf("\n please enter valid choice");

}
}
return 0;
}
```

OUTPUT:

```
 STACK USING ARRAY
.....MENU.....
 1.Push
 2.Pop
 3.Display
 4.Exit
.....
 Enter your choice1 Enter

 the element2

2 pushed in to the stack STACK
USING ARRAY
.....MENU.....
 1.Push
 2.Pop
 3.Display
 4.Exit
.....
 Enter your choice1 Enter the

 element98
```

98 pushed in to the stack STACK
USING ARRAY
.....MENU.....
1.Push
2.Pop
3.Display
4.Exit
.....
Enter your choice1

Enter the element7

7 pushed in to the stack
STACK USING ARRAY
.....MENU.....
1.Push
2.Pop
3.Display
4.Exit
.....
Enter your choice3

The stack elements are:2 98 7 STACK
USING ARRAY
.....MENU.....
1.Push
2.Pop
3.Display
4.Exit
.....
Enter your choice2

The value 7 poped from the stack

8. Implementation of Queue using array

```
#include <stdio.h>
#include<stdlib.h>
int SIZE = 10;
```

```
void insert(int a[], int *front, int *rear)
{
 if(*rear > SIZE - 2)
 {
 printf("\n Queue is full\n");
 return;
 }
 else
 {
 int e;
 printf("\n Enter a value\n");
 scanf("%d",&e);
 if(*front == -1 && *rear == -1)
 *front = *rear = 0;
 else
 *rear += 1;
 a[*rear] = e;
 printf("\n Entered element %d is inserted\n", e);  }
}
void delete(int a[], int *front, int *rear) {
 int e;
 if(*front < 0)
 {
 printf("\n Queue is empty\n");
 }
 else

 {
 e = a[*front];
 printf("\n Element %d is deleted\n",e);
 *front += 1;
 }
}
void display(int a[], int *front, int *rear)
{
 int i;
 if(*front < 0)
 {
 printf("\n queue is empty\n");
```

```
 return;
 }
 else
 {
 printf("\n The queue elements are:\n");
 for(i = *front;i <= *rear;i++)
 printf("%d ",a[i]);
 }
}
int main()
{
 int arr[SIZE];
 int front, rear;
 int ch, e=1;
 front = rear = -1;

 while(e)
 {
 printf("\n QUEUE USING ARRAY");
 printf("\n.....MENU.....");
 printf("\n 1.INSERT \n 2.DELETE \n 3.DISPLAY \n
4.EXIT");  printf("\n.....");
 printf("\n Enter your choice");
 scanf("%d",&ch);
 switch(ch)
 {
        case 1:
                insert(arr, &front, &rear);
                break;
        case 2:
                delete(arr, &front, &rear);
                break;
        case 3:
                display(arr, &front, &rear);
                break;


        case 4:
                e=0;
```

```
                printf("\n exiting...");
                break;
        default: printf("\n please enter valid choice");  }
 }

 return 0;
}
```

OUTPUT:

```
 QUEUE USING ARRAY
.....MENU.....
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
.....
 Enter your choice1

 Enter a value
12

 Entered element 12 is inserted

 QUEUE USING ARRAY
.....MENU.....
 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
.....
 Enter your choice1

 Enter a value
7

 Entered element 7 is inserted

 QUEUE USING ARRAY
.....MENU.....
```

1.INSERT
2.DELETE
3.DISPLAY

4.EXIT
.....
Enter your choice1

Enter a value
3

Entered element 3 is inserted

QUEUE USING ARRAY
.....MENU.....
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
.....
Enter your choice3

The queue elements are:
12 7 3
QUEUE USING ARRAY
.....MENU.....
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
.....
Enter your choice2

Element 12 is deleted

9.Imlementation of Circular Queue

```
#include <stdio.h>
```

```
#define SIZE 5

void enqueue(int a[], int *front, int *rear)
{
 int e;
 printf("\nEnter number:");
 scanf("%d", &e);
 if ((*rear + 1) % SIZE == *front)
 {
 printf("\nQUEUE overflow");
 return;
 }
 else if (*front > 0 && *rear == SIZE - 1)
 {
 *rear = 0;
 }
 else if ((*front == -1) && (*rear == -1))
 {
 *front = 0;
 *rear = 0;
 }
 else
 {
 printf("then");
 *rear += 1;
 }
 a[*rear] = e;
 printf("\nThe entered element %d is inserted in to
the  QUEUE\n", e);
}

void dequeue(int a[], int *front, int *rear)
{
 if (*front == -1)
 {
 printf("\nQUEUE underflow\n");
 }
 else if (*front == SIZE - 1)
 {
```

```
 *front = 0;
 }
 else


 {
 int e;
 e = a[*front];
 printf("\nThe element %d deleted from QUEUE",
e);  *front += 1;
 }
}

void display(int a[], int *front, int *rear)
{
 if (((*front == -1) && (*rear == -1)))
 {
 printf("Queue is empty");
 }
 else
 {
 int i;
 printf("\nthe QUEUE elements are:");
 if(*front>*rear)
 {
 for (i = *front; i<=(*rear+SIZE) ;
i++)  printf("\t%d", a[i%SIZE]);
 }
 else{
 for (i = *front; i<=(*rear) ; i++)
 printf("\t%d", a[i]);
 }

 }
}

void search(int a[], int *front, int *rear,int ele) {
 if (((*front == -1) && (*rear == -1)))
 {
```

```
printf("Queue is empty");
}
else
{
if(*front>*rear)
{
for (int i=*front;i<=(*rear+SIZE);i++)
{
if(a[i%SIZE]==ele)
{
printf("Item found!!!");


return;
}

}
printf("Item not found!!!");
}
else
{
for (int i=*front;i<=(*rear);i++)
{
if(a[i]==ele)
{
printf("Item found!!!");
return;
}

}
printf("Item not found!!!");
}

}
}

int main()
{
```

```
 int arr[SIZE], front = -1, rear = -1, ch, e = 1,
val;  while (e)
 {
 printf("\nCIRCULAR QUEUE OPERATIONS");
printf("\n_____MENU_____
_____\n");   printf("\n\t  1.  insert\n\t  2.  delete\n\t  3.
Display\n\t  4. Search\n\t 5. Exit\n");

printf("\n_____
_____\n") ;
 printf("\nEnter your choice:");
 scanf("%d", &ch);
 switch (ch)
 {
 case 1:
 enqueue(arr, &front, &rear);
 break;
 case 2:
 dequeue(arr, &front, &rear);
 break;


 case 3:
 display(arr, &front, &rear);
 break;
 case 4:
 printf("\nEnter the data to be
searched:");  scanf("%d", &val);
 search(arr, &front, &rear, val);
 break;
 case 5:
 e = 0;
 printf("\nExiting from the programe");  break;
 default:
 printf("\n please enter valid choice");  }
 }
 return 0;
}
```

OUTPUT:

CIRCULAR QUEUE OPERATIONS
_____MENU_____

1. insert
2. delete
3. Display
4. Search
5. Exit

_____

_____ Enter your choice:1

Enter number:2

The entered element 2 is inserted in to the QUEUE

CIRCULAR QUEUE OPERATIONS
_____MENU_____

1. insert
2. delete
3. Display
4. Search

5. Exit

_____

_____ Enter your choice:1

Enter number:4
then
The entered element 4 is inserted in to the QUEUE

CIRCULAR QUEUE OPERATIONS

_____MENU_____

1. insert
2. delete
3. Display
4. Search
5. Exit

_____

_____ Enter your choice:1

Enter number:6
then
The entered element 6 is inserted in to the QUEUE

CIRCULAR QUEUE OPERATIONS
_____MENU_____

1. insert
2. delete
3. Display
4. Search
5. Exit

_____

_____ Enter your choice:4

Enter the data to be searched:6
Item found!!!
CIRCULAR QUEUE OPERATIONS
_____MENU_____

_____ 1. insert

2. delete
3. Display

4. Search
5. Exit

_____

_____ Enter your choice:2

The element 2 deleted from QUEUE
CIRCULAR QUEUE OPERATIONS
_____MENU_____

1. insert
2. delete
3. Display
4. Search
5. Exit

_____

_____ Enter your choice:3

the QUEUE elements are: 4 6
CIRCULAR QUEUE OPERATIONS

10.Implementation of Queue using stack

```
#include <stdio.h>
#include<stdlib.h>
struct node
{
 int data;
 struct node *next;
};
struct node *front = NULL;
struct node *rear = NULL;

void insert()
```

```
{
 struct node *temp;
 int val;
 temp = (struct node*)malloc(sizeof(struct
node));  if(temp == NULL)
 {
 printf("\n Queue Overflow");
 return;
 }
 else
 {
 printf("\n Enter the value");
 scanf("%d",&val);
 temp -> data = val;
 temp -> next = NULL;
 if(front == NULL)
 front = rear = temp;
 else
 {
 rear -> next = temp;
 rear = temp;
 }
 printf("\n One value is inserted into the queue");  }


}
void delete()
{
 struct node *temp;
 if(front == NULL)
 {
 printf("\n Underflow");
 return;
 }
 else
 {
 temp = front;
 front = front -> next;
```

```
 printf("\n %d is deleted from the queue", temp ->
data);  free(temp);

 }
}
void display()
{
 struct node *temp;
 temp = front;
 if(front == NULL)
 {
 printf("\n Empty Queue");
 return;
 }
 else
 {
 printf("\n Queue elements are");
 while(temp != NULL)
 {
 printf("%d ", temp -> data);
 temp = temp -> next;
 }
 }
}
int main()
{
 int ch, e=1;

 while(e)
 {
 printf("\n QUEUE USING LINKED
LIST");  printf("\n.....MENU.....");
 printf("\n 1.INSERT \n 2.DELETE \n 3.DISPLAY \n
4.EXIT");  printf("\n.....");



 printf("\n Enter your choice");
 scanf("%d",&ch);
 switch(ch)
```

```
{
        case 1:
                insert();
                break;
        case 2:
                delete();
                break;
        case 3:
                display();
                break;
        case 4:
                e=0;
                printf("\n exiting...");
                break;
        default: printf("\n please enter valid choice");  }
}

 return 0;
}
```

OUTPUT:

```
QUEUE using STACKS

----------MENU----------

 1. Enqueue
 2. Dequeue
 3. Display
 4. Exit

-----------------------
Enter your choice:1
Enter the element:23

----------MENU----------

 1. Enqueue
 2. Dequeue
 3. Display
```

```
 4. Exit

-----------------------



Enter your choice:1
Enter the element:6

----------MENU----------

 1. Enqueue
 2. Dequeue
 3. Display
 4. Exit

-----------------------
Enter your choice:1
Enter the element:9

----------MENU----------

 1. Enqueue
 2. Dequeue
 3. Display
 4. Exit

-----------------------
Enter your choice:3
 23 6 9
----------MENU----------

 1. Enqueue
 2. Dequeue
 3. Display

 4. Exit

-----------------------
Enter your choice:2
The element 23 is deleted from queue
```

11. Write a program to find the sum of all elements in an array which are at odd position

```
#include<stdio.h>
#include<conio.h>
int sum(int [],int);
int main()
{
int a[20],n,i;
printf("Enter the range:");
scanf("%d",&n);
printf("Enter the elements\n");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
printf("Required sum:%d",sum(a,n));
getch();
return 0;
}
int sum(int a[],int n)
{
int i,s=0;
for(i=0;i<n;i++)
{
if(i%2!=0)
s=s+a[i];
}
```

```
return s;
}
```

OUTPUT:

```
Enter the range:5
Enter the elements
1
2
3
4
5
Required sum:6
```

12.sum and difference of matrix

```
#include<stdio.h>

int main()
{
   int n, m, c, d, first[10][10], second[10][10], sum[10][10], diff[10][10];
   printf("\nEnter the number of rows and columns of the first matrix \n\n");
   scanf("%d%d", &m, &n);

   printf("\nEnter the %d elements of the first matrix \n\n", m*n);
   for(c = 0; c < m; c++)   // to iterate the rows
```

```
   for(d = 0; d < n; d++)   // to iterate the columns
      scanf("%d", &first[c][d]);

printf("\nEnter the %d elements of the second matrix \n\n", m*n);
for(c = 0; c < m; c++)   // to iterate the rows
   for(d = 0; d < n; d++)   // to iterate the columns
      scanf("%d", &second[c][d]);
printf("\n\nThe first matrix is: \n\n");
for(c = 0; c < m; c++)   // to iterate the rows
{
   for(d = 0; d < n; d++)   // to iterate the columns
   {
      printf("%d\t", first[c][d]);
   }
printf("\n");
}
printf("\n\nThe second matrix is: \n\n");
for(c = 0; c < m; c++)   // to iterate the rows
{
   for(d = 0; d < n; d++)   // to iterate the columns
   {
      printf("%d\t", second[c][d]);
   }
printf("\n");
}
```

```
   for(c = 0; c < m; c++)
      for(d = 0; d < n; d++)
         sum[c][d] = first[c][d] + second[c][d];
   printf("\n\nThe sum of the two entered matrices is: \n\n");
   for(c = 0; c < m; c++)
   {
      for(d = 0; d < n; d++)
      {
         printf("%d\t", sum[c][d]);
      }
      printf("\n");
   }

   for(c = 0; c < m; c++)
      for(d = 0; d < n; d++)
         diff[c][d] = first[c][d] - second[c][d];

   printf("\n\nThe difference(subtraction) of the two entered matrices is:
\n\n");
   for(c = 0; c < m; c++)
   {
      for(d = 0; d < n; d++)
      {
         printf("%d\t", diff[c][d]);
      }
      printf("\n");
   }
```

```
    return 0;
}
```

OUTPUT:

```
Enter the number of rows and columns of the first matrix 3
3
Enter the 9 elements of the first matrix
1 4 3 6 7 408 4
Enter the 9 elements of the second matrix
2 4 5 9 6 7 0 3 4
The first matrix is
478
The second matrix is
The sum of the two entered matrices is :
3
15
0
8
13
11
8 11 8
The difference <subtraction) of the two entered matrices is :
-2
-3
-3
0
```

13.Find out the Transpose of a matrix

```
#include <stdio.h>
int main() {
  int a[10][10], transpose[10][10], r, c;
  printf("Enter rows and columns: ");
  scanf("%d %d", &r, &c);

  printf("\nEnter matrix elements:\n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }
  printf("\nEntered matrix: \n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("%d  ", a[i][j]);
    if (j == c - 1)
    printf("\n");
  }
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j];
  }
```

```
  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j) {
    printf("%d  ", transpose[i][j]);
    if (j == r - 1)
    printf("\n");
  }
  return 0;
}
```

OUTPUT:

```
Enter rows and columns: 2
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 0
Enter element a21: -5
Enter element a22: 2
Enter element a23: 7
```

Entered matrix:

1 4 0

-5 2 7

Transpose of the matrix:

1 -5

4 2

0 7

14.fibanacci up to a range

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c,n;
clrscr();
printf("\nEnter range:");
scanf("%d",&n);
a=0,b=1,c=0;
printf("%d \t %d",a,b);
c=a+b;
while(c<=n)
{
printf("\t%d",c);
a=b;
```

```
b=c;
c=a+b;
}
getch();
}
```

OUTPUT:

```
Enter   range:13
0       1       1       2       3       5       8       13
```

15. Execute Matrix multiplication

```
#include <stdio.h>
void main()
{
int a[25][25],b[25][25],c[25][25],i,j,k,r,s;
int m,n;
printf("Enter the first matrix\n");
scanf("%d%d",&m,&n);
printf("Enter the second matrix\n");
scanf("%d%d",&r,&s);
if(m!=r)
printf("\n The matrix cannot multiplied");
else
{
printf("\n Enter the elements of first matrix ");
```

```
for(i= 0;i<m;i++)
{
for(j=0;j<n;j++)
scanf("\t%d",&a[i][j]);
}
printf("\n Enetr the elements of second matrix ");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
scanf("\t%d",&b[i][j]);
}
printf("\n The element of first matrix is");
for(i=0;i<m;i++)
{
printf("\n");
for(j=0;j<n;j++)
printf("\t%d",a[i][j]);
}
printf("\n The element of second matrix is");
for(i=0;i<m;i++)
{
printf("\n");
for(j=0;j<n;j++)
printf("\t%d",b[i][j]);
}
for(i=0;i<m;i++)
{
```

```
printf("\n");
for(j=0;j<n;j++)
{
c[i][j]=0;
for(k=0;k<m;k++)
c[i][j]=c[i][j]+a[i][k]*b[k][j];
}
}
}
printf("\n Multiplication of two matrix is");
for(i=0;i<m;i++)
{
printf("\n");
for(j=0;j<n;j++)
printf("\t%d",c[i][j]);
}
}
```

OUTPUT:

```
Enter the first matrix
2  2
Enter the second matrix
2  2
Enter the elements of first matrix
```

1 2

3 4

Enetr the elements of second matrix 1 2 1 2

The element of first matrix is

1 2

3 4

The element of second matrix is

1 2

1 2

Multiplication of two matrix is

3 6

7 14