LAB IX

1.
```
#include<stdio.h>
void reverse(char *);
int check_char(char*,int);
int main()
{
int flag=0;
int c;
char p[]="Hellow world";
char *q="MTH409: C program";

printf("First string is :%s\n",p);
reverse(p);
printf("First string after function call is :%s\n",p);
printf("Enter the search character:");
c=getchar();
flag=check_char(q,c);
if(flag==0)
{
        printf("The character %c is not present in \"%s\"\n",c,q);
}
else
{
                printf("The character %c is present in \"%s\"\n",c,q);
}

return 0;
}
void reverse(char *a)
{
        int i,n=0,t,c;
        while(a[n] !='\0')
        {
                n++;
        }
        for(i=0;i<n;i++,n--)
        {
                t=a[i];
                a[i]=a[n-1];
                a[n-1]=t;
        }
}

int check_char(char *a,int c)
{

        while(*a !='\0')
        {
                if(c==*a)
                {
```

```
                        return 1;
                }
                a++;
        }

return 0;
}
```

2.
```
/* This prog. reads & prints a polynomial*/
#include <stdio.h>
#include <stdlib.h>
void polprn(int,double *);
double *polc(int *,char *);
int main()
{
  int n;
  double *a;
   a =polc(&n,"poly.dat");
   polprn(n,a);
return 0;
}


double *polc(int *m,char *filename)
{
  int i,n;
  FILE *inpol;
  double *a;
  inpol=fopen(filename,"r");
  if(inpol==NULL)
  {printf("Error in opening %s\n",filename);
   exit(1);
  }

  fscanf(inpol,"%d",&n);
  if(n<0){
      printf("Degree must be non-negative\n");
      exit(1);
  }

  a=(double *)calloc(n+1,sizeof(double));
  if(a==NULL){
       printf("Error in allocation\n");
       exit(1);
  }
  for(i=n;i>=0;i--)
     fscanf(inpol,"%lf",&a[i]);
```

```
  *m=n;
  return a;
}

void polprn(int n,double *a)
{
int i;
printf("The polynomial is :");
if(n==0){
printf("%0.2lf \n",a[n]);
return;
}
printf("%0.2lf x^%d",a[n],n);
for(i=n-1;i>=0;i--)
{

 if(a[i]!=0)
  {
          printf("%+0.2lf ",a[i]);
          if(i!=0)
             {
                  printf("x^%d",i);
             }
  }

}

printf("\n");
}
```

3.
```
#include <stdio.h>
#include <stdlib.h>
double **matr(int *,int *,char *);
void matrr(int *,int *,double ***,char *);
void matprin(char,double **,int,int);
int main()
{
int m,n;
double **A;

A=matr(&m,&n,"matA.dat");
//Following void function also do the same thing
//matrr(&m,&n,&A,"matA.dat");
matprin('A',A,m,n);
return 0;
}

void matprin(char mat,double **A,int m,int n)
{
int i,j;
printf("The matrix %c is:\n",mat);
```

```
for(i=0;i<m;i++)

        {
        for(j=0;j<n;j++)
        {
                printf("%0.3lf  ",A[i][j]);
        }
 printf("\n");
}

}


void matrr(int *m,int *n,double ***A,char *fnm)
{
int p,q,i,j;
double **a;
FILE *fp;
fp = fopen(fnm,"r");
if(fp==NULL){
printf("Error in opening %s\n",fnm);
exit(1);}

fscanf(fp,"%d%d",&p,&q);
if(p<=0 || q<=0)
{
        printf("Matrix dimensions must be positive\n");
        exit(1);
}
*m=p;
*n=q;

a=(double **)calloc(p,sizeof(double *));
*A = a;
if(a==NULL)
{
printf("Error in calloc\n");
exit(1);
}

for(i=0;i<p;i++)
 {
   a[i]=(double *)calloc(q,sizeof(double));
   if(a[i]==NULL){
          printf("Error in calloc\n");
          exit(1);}
 }

for(i=0;i<p;i++)
for(j=0;j<q;j++)
fscanf(fp,"%lf",&a[i][j]);
```

```
}


double **matr(int *m,int *n,char *fnm)
{
int p,q,i,j;
double **a;
FILE *fp;
fp = fopen(fnm,"r");
if(fp==NULL){
printf("Error in opening %s\n",fnm);
exit(1);}

fscanf(fp,"%d%d",&p,&q);
if(p<=0 || q<=0)
{
        printf("Matrix dimensions must be positive\n");
        exit(1);
}
*m=p;
*n=q;

a=(double **)calloc(p,sizeof(double *));
if(a==NULL)
{
printf("Error in calloc\n");
exit(1);
}

for(i=0;i<p;i++)
 {
   a[i]=(double *)calloc(q,sizeof(double));
   if(a[i]==NULL){
          printf("Error in calloc\n");
          exit(1);}
 }

for(i=0;i<p;i++)
for(j=0;j<q;j++)
fscanf(fp,"%lf",&a[i][j]);
return a;
}
```