```c
clrscr();
      printf("\n\t\t This program needs Three files....\n\n");
      printf("\n\t\t\t One : INPUT.DAT -> Input Data of dim I x 1\n");
      printf("\n\t\t\t Two : D_OUT.DAT -> Target data of dim K x 1\n");
      printf("\n\t\t\t Three :TEST.dat -> Test data of dim I x 1\n");


      printf("\n\t\t If Files exit, then To Continue, Press <Enter>\n");
          c= getche();

        if(c == '')
            exit(0);

          clrscr();
      /* enter the data....*/

      printf("\n\tEnter the values of\n\tNo.of Neurons= I");
      printf("\n\tDim of Target vector= K ");
      printf("\n\tNo. of Pattern vectors= P\n");
          scanf("%d%d%d",&I,&K,&P);
                I=I+1;
                /* Reading the values of Input from File */
                /*       .DAT */
                  for (p=1;p<=P;p++)
                        {
                        for(i=1;i<=I-1;i++)
                          fscanf(fp1,"%f",&z[p][i]);
                             z[p][I]=-1;
                        }
                printf("\n");

                      for (p=1;p<=P;p++)
                        for(k=1;k<=K;k++)
                             fscanf(fp2,"%f",&d[p][k]);

      printf("\n\tEnter the value of \n\tIta and Emax & qmax\n ");
       scanf(" %f%f%d",&ita,&emax,&qmax);


          /* generate  v[] */
            randomize();
                for(k=1;k<=K;k++)
                 for(i=1;i<=I;i++)
                  v[k][i]=2.0*((float)rand()/RAND_MAX)-1.0;


          /*      iteration starts */

                q=0;
      while( q < qmax)
      {
                e=0;q=q+1;
```

```
                    randomize();
                     r1= random(P);
                        if(r1==0)
                          p=1;
                        else
                         p = r1;
                        ip=1;


                  while(ip<= P)

          {
                        /* transfer z[][] to z1[],d[][] to d1[]*/

                        for(i=1;i<=I;i++)
                         z1[i]=z[p][i];

                        for(k=1;k<=K;k++)
                        {
                         d1[k]=d[p][k];
                         }

/* compute net*/
          for(k=1;k<=K;k++)
        {
                  s=0;
              for(i=1;i<=I;i++)
              s=s+v[k][i]*z1[i];
                        m[k]=s;
            }

/* compute f(net)*/
          for (k=1;k<=K;k++)
          y[k]= (2/(1.0+exp(-m[k])))-1;


/* error computation*/
            for(k=1;k<=K;k++)
                e=e+0.5*(d1[k]-y[k])*(d1[k]-y[k]);


/* error correction */
   for(k=1;k<=K;k++)
          ed[k]=(d1[k]-y[k])*(1-y[k]*y[k])/2;


   /* Updation of wts */
          for(k=1;k<=K;k++)
              for(i=1;i<=I;i++)
              v[k][i]=v[k][i]+ita*ed[k]*z1[i];
```

```c
                    /* input next z */

                    p=p+1;
                     if(p > P)
                        p=1;
                    ip=ip+1;

        }
                /* end of for loop for inputting all z. */

    printf("\n\t Error is = %.5f\t and Iteration no.= %d ",e,q);

                    if(( q>= qmax) || (e < emax))
        {

                    clrscr();
                printf("\n\n Process is over.....\n");
                printf("\n\t Estimated weights are ...\n");
                    for(k=1;k<=K;k++)
                        for(i=1;i<=I;i++)
    printf("\n\n\t   v[%d][%d] = %.5f",k,i,v[k][i]);

            e=sqrt(e)/P*K;
    printf("\n\n\t\t\t\tNo. of Iter.= %3d & Error is = %4.4f ",q,e);
                                break;
                        }

        } /* end of while loop(q<qmax) */


        /* For test data..... */

            printf(" \n\n\n\t Enter size of Test data ..\n");
                    scanf("%d",&T);
                for(t=1;t<=T;t++)
                    for(i=1;i<=I-1;i++)
                      {
                            fscanf(fp3,"%f",&z[t][i]);
                            z[t][I]=-1;
                      }

                for(t=1;t<=T;t++)
                 {
                    for(i=1;i<=I;i++)
                        z1[i]=z[t][i];

                /* Compute net output */

                    for(k=1;k<=K;k++)
                      {
                            s=0;
                            for(i=1;i<=I;i++)
                            s=s+v[k][i]*z1[i];
                            m[k]=s;
                      }
```

132

```c
           for(k=1;k<=K;k++)
             {
                 y[k]= (2/(1.0+exp(-m[k])))-1;
                 y1[t]=y[k];
             }
             } /* end of test values */

             s=0;
     for(t=1;t<=T;t++)
       {
                 for(k=1;k<=K;k++)
                 d1[t]=d[t][k];
                 printf("\n\n\t %.4f ",y1[t]);

                 pre= s+(y1[t]-d1[t])*(y1[t]-d1[t]);

fprintf(fp5,"\n the value of y1[%d]=%.4f,d[1]=%.4f",t,y1[t],d1[t]) ;
             }

             fclose(fp5);
             printf("\n Predictive error is= %f ",pre);
             printf("\n\n\n\t\t\t\t\t\ Program ends here.......");

}
```