# APPENDIX II (A)

## C-PROGRAM FOR SECTION [3.5]

## MINIMUM OF A WEIBULL DISTRIBUTION

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <ctype.h>
void main()
 {
   float d,d1,beta,alpha,f;
   double x[5000],y,z,min,a,b,c1,s2=0,s3=0,eb=0,p=0;
   int i,j=1,k=0,k1=0,k2=0,m=1,l,r,c,q;long s=0;long s1=0;long s4=0;
   FILE *fp;
   fp=fopen("web.xls","w+");
   fprintf(fp,"Observed values of alpha and median sample sizes for simulation
      studies with m=2");
   fprintf(fp,"\n BETA  D  ALPHA  SALPHA  MS-SIZE MSALPHA MMS-SIZE EBETA MLEA
      MLE-SIZE");
   clrscr();
   randomize();
   for(beta=.5;beta<=1.0;beta+=.5)
   { z=1.0/beta;
     for(d1=.1;d1<=.3;d1+=.1)
      { d=d1*(pow(log(10),z)-pow(log(10/9),z));
      for(alpha=.05;alpha<=.15;alpha+=.05)
      {
        j=1;i=1;s=0;k=0;l=0;s1=0;k1=0;s4=0;k2=0;m=1;p=0;
        for(i=1;i<=5000;i++)
        {
           for(j=1;j<=5000;j++)
           {
            x[j]=pow(-log(1.0-(float)random(32000)/32000),z);
            if(fabs(x[j])==0) {j=j-1;continue;};
           }

/* MECZARSKI PROCEDURE */
         min=x[1];j=1;l=0;
         do{ j+=1;
             for(m=1;m<=j;m++) if(x[m]<min) min=x[m];
             for(m=1;m<=j;m++)
             if(x[m]<=min+d) l+=1;
             f=(float)l/j;
             if(j>=log(alpha)/log(1.0-f))
            {s+=j;if(min-d<0 && 0<min) k+=1;goto a;}
             l=0;
            }while(j<5000);
          if(min-d<0 && 0<min) k+=1;goto b;
```

```
/* MODIFIED MECZARSKI PROCEDURE */
          a:l=0;q=j;c=9;r=0;
           do{ r+=1;
            for(m=q+1;m<=q+10;m++) if(x[m]<min) min=x[m];
            for(m=1;m<=q+10;m++)    if(x[m]<= min+d) l+=1;
            f=(float)l/(q+10);c1=q+10-log(alpha)/log(1.0-f);
            if(c1>=c)
             {s1+=q+10;break;}
            else {l=0;q=q+10;if(r<8) c=c+9-r;
                  else {if(r==8) c=48;else c=50;};continue;}
             }while(r<10);
           if(min-d<0 && 0<min) k1+=1;

/* M.L.E PROCEDURE */
        b: min=x[1];j=1;s2=0;m=1;eb=0;
          do{ j+=1;
            for(m=1;m<=j;m++) {if(x[m]<min) min=x[m];s2+=log(x[m]);};
            do{ eb+=0.0001;s3=0.0;
                for(m=1;m<=j;m++)
                 { a=pow(x[m],eb);
                   s3+=log(x[m])*a;
                 }
               }while(s2+j/eb-s3 > 0);
            b=pow(d+min,eb);
            if(j>=log(1.0/alpha)/b)
               {s4+=j;break;}
            s2=0;eb=0;
             }while(j<5000);
           if(min-d<0 && 0<min) k2+=1;p+=eb;printf("\n %d \t %f \t %d",i,eb,c);

         }

      fprintf(fp,"\n %2.2f %2.2f %4.3f %6.4f %4ld %6.4f %4ld %6.4f %6.4f
      %4ld",beta,d1,alpha,1.0-k/5000.0,s/5000,1.0-k1/5000.0,s1/5000,p/5000,1.0-
      k2/5000.0,s4/5000);
       }
     }
    }
getch();
fflush(stdin);
fclose(fp);
}

/* PROGRAM END */
```

APPENDIX II(B)
**********

C-PROGRAM FOR SECTION [3.5.1]

MINIMUM OF A PARETO DISTRIBUTION
**********

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <ctype.h>
void main()
 {
   float d,d1,z,theta,alpha,f,c1,p1,p2;double x[5000],y,min,t;
   int i,j,k=0,k1=0,m,n,l,k2=0,k3=0,q,r=0,c=0;long s=0;long s1=0;
   long s2=0;long s3=0;
   FILE *fp;
   clrscr();
   randomize();
   fp=fopen("p3.xls","w+");
   fprintf(fp,"\n THETA D ALPHA SALPHA MS-SIZE AVG-C MSALPHA MMS-SIZE EALPHA
      EMS-SIZE MEALPHA MEMS-SIZE");
   for(theta=0.5;theta<=2.0;theta+=.5) {z=1.0/theta;
   for(d1=.1;d1<=.5;d1+=.1) {d=d1*(pow(10,z)-pow(10/9,z));
   for(alpha=.001;alpha<=.1;alpha=alpha*10) {
   for(i=1;i<=5000;i++)
     {
       printf("\n %d",i);
       for(j=1;j<=5000;j++)
      { x[j]=1.0/pow(1.0-(float)random(32000)/32000,1.0/theta);
       if(fabs(x[j])==0) {j=j-1;continue;};
       }
/* MECZARSKI PORCEDURE */
       j=1;l=0;min=x[1];
      do{ j+=1;
        for(m=1;m<=j;m++) if(x[j]<min) min=x[j];
        for(m=1;m<=j;m++)
         if(x[m]<= min+d) l+=1;
        f=(float)l/j;
        if(j>-log(alpha)/log(1.0-f))
         { s+=j;if(min-d<1 && 1<min) k+=1;goto a;};l=0;
       }while(j<5000);
       if(min-d<1 && 1<min) k+=1;
       goto b;
```

```
 /* MODIFIED MECZARSKI PROCEDURE */
 a:    l=0;q=j;c=9;r=0;
       do{ r+=1;
           for(m=q+1;m<=q+10;m++) if(x[m]<min) min=x[m];
           for(m=1;m<=q+10;m++)   if(x[m]<= min+d) l+=1;
           f=(float)l/(q+10);c1=q+10-log(alpha)/log(1.0-f);
           if(c1>=c)
            {s1+=q+10;break;}
           else {l=0;q=q+10;if(r<=8) c=c+9-r;else c=48;continue;};
         }while(r<10);
       if(min-d<1 && 1<min) k1+=1;

/* USING MOMENT ESTIMATOR */
 b:     j=1;t=0;min=x[1];
       do{ j+=1;
           for(m=1;m<=j;m++) if(x[j]<min) min=x[j];
           for(m=1;m<=j;m++)
            t+=x[m];
           if(j>=log(1.0/alpha)/(t/(t-j)*log(min+d)))
             {s2+=j;p1+=t/(t-j);break;};
            t=0;
         }while(j<5000);
       if(min-d<1 && 1<min) k2+=1;

/* USING M.L.E. */
       j=1;t=0;min=x[1];
       do{ j+=1;
           for(m=1;m<=j;m++) if(x[j]<min) min=x[j];
           for(m=1;m<=j;m++)
             t+=log(x[m]);
           if(j>=log(1.0/alpha)/(j/t*log(min+d)))
             {s3+=j;p2+=j/t;break;};
           t=0;
          }while(j<5000);
       if(min-d<1 && 1<min) k3+=1;
     }

    fprintf(fp,"\n %2.2f %2.2f %5.4f %6.4f %4ld %6.4f %4ld %6.4f
    %6.4f %4ld %6.4f %6.4f %4ld",theta,d1,alpha,1.0-k/5000.0,s/5000,1.0-
      k1/5000.0,s1/5000,p1/5000,
      1.0-k2/5000.0,s2/5000,p2/5000,1.0-k3/5000.0,s3/5000);
    s=0;k=0;i=1;m=1;k1=0;j=1;s1=0;k2=0;s2=0;k3=0;s3=0;c=0;p1=0;p2=0;
   }
  }
 }
   getch();
   fflush(stdin);
   fclose(fp);
 }

/* PROGRAM END */
```