APPENDIX 1

<u>Program No.</u> : 1.1

<u>Purpose</u> : To obtain ARL values for Normal Distribution by simulation.

Note : **To generate single observation from SNV, we sum 12 independent uniform variate over (0, 1) and subtract 6 from it.**

```
# include <stdio.h>
# include <math.h>
# include <stdlib.h>
# include <time.h>
# include <dos.h>
# include <conio.h>
void main( )
{ float a, s, m, arl1, v, w, z, d;
  long sum;
  int i, j, k, l, n = 10;
  clrscr( );
  randomize( );
  printf("\n          ARL Values of L(+) for Shewhart Chart using ");
  printf("\n          x-bar when n=10 for shift in Normal Dist. N(0,1) ");
  printf("\n  =====================================");
  printf(" \n            m - mo  |   ARL");
  printf(" \n          ——————————————");
    for(d=0.0; d<=2.0; d = d + 0.25)
        { printf("\n          %6.2f   |" ,d);
          sum=0.0;
              for(i=1;i<=1000; i=i+1)
                { m=0.0; j=0;
                  do
                   { j = j + 1; s = 0.0;
                     for (k=1;k<=n;k++)
                         { z=0.0;
                           for (l=1;l<=6;l++)
                            { v=(float) random(32000)/32000;
                              w=(float) random(32000)/32000;
                              z=z+v+w;
                            } /* enf of loop l */
                           z=(z-6)+d;
                           s=s+z;
                         } /* enf of loop k */
                     m=(float) s/n;
                   }while(fabs(m)<0.97818);
                    sum=sum+j;
                 } /* enf of loop i */
                   arl1=(float) sum/(i-1);
                   printf(" \a   %5.1f", arl1);
        } /* end of loop d */
   printf(" \n          ——————————————————");
   printf(" \n           The number of iterations = %d ",i-1);
   printf(" \n          ——————————————————");
   getch();
 }
```

Program No. : **1.2**

Purpose : To obtain ARL values for Uniform Distribution by simulation.

Note : Uniform Distribution : U($-\sqrt{3}$, $\sqrt{3}$)

Gives Mean = 0 & Var = 1 ; X = $\sqrt{3}$ (2*U – 1), where U → U(0, 1)

```
# include <stdio.h>
# include <math.h>
# include <stdlib.h>
# include <time.h>
# include <dos.h>
# include <conio.h>
void main()
{ float a, sum , s,m,arl1;
  float v, w, z, d;
  int i, j, k, l;
  clrscr();
  randomize();
  printf("\n          ARL Values of L(+) for Shewhart Chart using ");
  printf("\n          x-bar when n=10 for shift in Uniform U(0,1) ");
  printf("\n  ==========================================");
  printf(" \n            m - mo   |    ARL");
  printf(" \n          ------------------------");
    for(d=0.0;d<=2.0;d=d+0.25)
      { printf("\n                 %6.2f    |" ,d);
        sum=0.0;
         for(i=1;i<=10; i=i+1)
           {m=0.0;j=0;
             do
              {j=j+1; s=0.0;
                for (k=1;k<=10;k++)
                 { v=(float) random(32000)/32000;
                   z=sqrt(3)*(2*v-1)+d;
                   s=s+z;
                 } /* end of loop k */
                   m=s/10.0;
              }while(m<0.948683);
               sum=sum+j;
           }  /* end of loop  i */
         arl1=(float)sum/(i-1);
         printf(" \a   %5.1f", arl1);
      } /* end of loop d */
   printf(" \n            -------------------------------");
   printf(" \n        The number of iterations = %d ",i-1);
   printf(" \n            -------------------------------");
   getch();
 }
```

Program No. 1.3

Purpose : To obtain ARL values for Gamma Distribution, by simulation.

Note : For Gamma Distribution : G(K, θ) if K=4 & θ=2,
Gives Mean=2 & Var=1; Generate Exponential variate,E -> -(1/2)logU ;
Sum 4E's for 1 Gamma variate with mean=2, where U → U(0, 1).

```
 # include <stdio.h>
# include <math.h>
# include <stdlib.h>
# include <time.h>
# include <dos.h>
# include <conio.h>
void main()
  {float a, sum, s, m, arl1, z, d, g, e, u1;
   int i,j,k,l;
clrscr();
randomize();
printf("\n        ARL Values of L(+) for Shewhart Chart using ");
printf("\n      x-bar when n=10 for shift in Gamma Dist. G(4, 2) ");
printf("\n   ==========================================");
 printf("\n              m - mo  |    ARL");
 printf("\n              ----------------------");
     for(d=0.0; d<= 2.0 ; d = d + 0.25)
       { sum=0.0;z=0;
         for(i=1; i<=1000; i=i+1)
           { m=0.0;j=0;
             do
               {j=j+1; s=0.0;
                 for (k=1;k<=10;k++)
                   { g=0;
                     for (l=1;l<=4;l++)
                       { u1=(float) random(32000)/32000;
                         if (u1==0)
                             {continue;}
                        else
                             { e=-0.50*log(u1);}
                       g=g+e;
                     } /* end of loop l */
                       g=g+d; s=s+g;
                   } /* end of loop k */
                   m=s/10.0-2.0;/* Mean is shifted to zero */
               }while(m<0.948683);
             sum=sum+j;
           } /* end of loop i */
     arl1=(float)sum/(i-1);
     printf("\n                    %4.2f   |   %5.1f",d, arl1);
   } /* end of loop d */
     printf(" \n          -------------------------------");
     printf(" \n     The number of iterations = %d ",i-1);
     printf(" \n          -------------------------------");
     getchar();
}
```

Program No. : **1.4**

Purpose : To obtain ARL values for Double Exponential Distribution by simulation.

Note : For Double exponential Distribution : $f(x) = 1/\sqrt{2} * \exp(-\sqrt{2} * |x|)$,
Gives Mean = 0 & Var = 1;

```
# include <stdio.h>
# include <math.h>
# include <stdlib.h>
# include <time.h>
# include <dos.h>
# include <conio.h>
 void main()
    {float a, sum, s, m, arl1, z, d, root2=sqrt(2);
     long double u1;
     int i,j,k,l;
 clrscr();
 randomize();
 printf("\n    ARL Values of L(+) for Shewhart Chart using ");
 printf("\n   x-bar when n=10 for shift in Double Exponential Dist.");
 printf("\n  ========================================");
 printf("\n              m - mo  |    ARL");
 printf("\n           -------------------------");
  for(d=0.0; d<=2.0; d = d + 0.25)
    { sum=0.0;z=0;
       for(i=1; i<=5000; i=i+1)
         { m=0.0; j=0;
           do
            {j=j+1;s=0.0;
              for (k=1;k<=10;k++)
                { u1=(float) random(32000)/32000;
                  if (u1<0.5)
                     { if (u1==0) continue;
                       else
                       z=log(2*u1)/root2;}
                   else
                     {if (u1==1)  continue;
                      z=-log(2*(1-u1))/root2;}
                      z=z+d; s=s+z;
                } /* end of loop k */
                  m=s/10.0;
            } while(fabs(m)<1.045);
              sum=sum+j;
        } /*  end of loop i  */
         arl1=(float)sum/(i-1);
         printf("\n                    %4.2f   |    %5.1f",d, arl1);
   } /* end of loop d */
   printf(" \n           ---------------------------------");
   printf(" \n             The number of iterations = %d ",i-1);
   printf(" \n           --------------------------------");
   getchar();
}
```

Program No. : **1.5**

Purpose : To obtain ARL values for Exponential Distribution by simulation.

Note : For Exponential Distribution : f(x)= exp(-x) , Gives Mean=1 & Var=1
Exponential E -> -1*log(U) ; with mean=1. where U ~ U(0,1)

```
# include <stdio.h>
# include <math.h>
# include <stdlib.h>
# include <time.h>
# include <dos.h>
# include <conio.h>
 void main()
   {float a, sum, s, m, arl1, z, d, g, e, u1;
    int i,j,k,l,n=10;
 clrscr();
 randomize();
 printf("\n        ARL Values of L(+) for Shewhart Chart using ");
 printf("\n        x-bar when n=10 for shift in Exponential Dist.");
 printf("\n  ====================================");
 printf("\n              m - mo   |    ARL");
 printf("\n            -------------------------");
 for(d=0.0; d<= 2.0; d=d+0.25)
   { sum=0.0; z=0;
    for(i=1; i<=10; i=i+1)
         { m=0.0; j = 0;
          do
            { j=j+1; s=0.0;
             for (k=1;k<=n;k++)
               { u1=(float) random(32000)/32000;
                 { if (u1==0) continue;
                   else
                     e=-1*log(u1)+d;
                 }
                 s=s+e;
               } /* end of loop k */
             m=(float) s/n-1.0;/* Mean is shifted to zero */
            }while(m<0.948683);
          sum=sum+j;
         } /* end of loop i */
    arl1=(float)sum/(i-1);
    printf("\n                  %4.2f   |   %5.1f",d, arl1);
   } /* end of loop d */
  printf(" \n          --------------------------------");
  printf(" \n           The number of iterations = %d ",i-1);
  printf(" \n          --------------------------------");
  getchar();
 }
```

Program No. : **2.1**

Purpose : To obtain in-control ARL values for Normal Distribution by simulation

SHEWHART CONTROL CHARTS USING THE SIGN STATISTICS

/* Ref. Amin, Reynold and Bakir (1995) :1597-1623 */

```
# include <stdio.h>
# include <math.h>
# include <stdlib.h>
# include <time.h>
# include <dos.h>
# include <conio.h>
  void main()
{
   float arl, z, v, w, x1;
   long int rl;
   int i,j,k,l,t,sn,w2,r,c;
   clrscr ( );
   printf(" \n   L(+) Values for Shewhart Charts Using the Sign Statistics   ");
   printf( "\n              when n = 10 and  a2 = 8               ");
   randomize();
   for(r=2;r<=7;r=r+1)
     { printf("\n r=%d  |",r);
        for(w2=0;w2<=6;w2=w2+2)
         { rl=0;
           for(i=1;i<=5000;i++)
             { t=0; c=100;
               for(k=1;k<=c;k++)
                 { sn=0;
                   for(j=1;j<=10;j++)
                     { z=0;
        /*  The random numbers generated from N(0,1)
                         for(l=1;l<=6;l++)
                           { v=(float) random(32000)/32000;
                             w=(float) random(32000)/32000;
                             z=z+v+w;
                           }  /* end of loop l */
                         x1=(z-6);*/
                         if (x1>0)
                             {sn=sn+1;}
                        else
                             {sn=sn-1;}
                     }/* end of loop j */
                   if (sn<8)  /* Value of a2  */
                     { if (sn>=w2) /* value of w2  */
                         { t=t+1; c=c+1;
                           if (t==r) /* value of r  */
                             {t=0;break;} /* r succ. points in WCL & UCL
                                               then process is out of control*/
                           else
                             {continue;} /* sn is between WCL & UCL */
                         }
                        else
                           {c=c+1; t=0; continue;} /* sn > UCL, process out of control*/
                     }
                   else
                    {break;} /* if sn<warning limit, then continue*/
```

```
          } /* end of loop for k  */
           rl=rl+k;
         }  /* end of for i */
          arl=(float) rl/(i-1); rl=0;
          printf(" :  %6.1f  ",arl);
        }  /* end of loop for w2 */
          printf("\n      |   ");
      }  /* end of loop for r */
  printf(" \n  =========================================================");
  printf(" \n              No. of iterations = %d ",(i-1));
  printf("\n  Note : The random numbers are generated from N(0,1)");
  getchar();
}
```