REFERENCES

1. James, J. Higgins, Sallle – Kellr – MeNulty "Concept in probability and Stochastic Modeling", 1995.
2. Macmillan Publishing Company, Collier Macmillan Canda, Maxwell Macmillan International "Data and Computer Communications", 1991.
3. S. Hillier, Frederick, J. Lieberman, Gerald, “Introduction to Operation Research” 1995.
4. Jerry Banks, Johns. Carson II, Barry L. Nelson, David M. Nicol, "Discrete-Even System Simulation", prentice-Hall of India private limited, New Delhi-2003.
5. James S. Dyer, Elbnood S. Buff, "Model Formula and Solution Methods" 1977, 1981.
6. Hamdy, A. Taha, "Operations Research: An Introduction" Prentice-Hall of India Private Limited, 2002.
7. S. French, R. Hartely, L. C. Thomas and D. J. White. "Operational Research Techniques" Edward Arnold.
8. Olivier Lecarme, Jean-Louis Nebut, "Pascal for Programmer" Mc Graw – Hill International Editions” 1986.
9. Richard L. Mills, "Statistics For Applied Economics and Business" 1977.
10. Ross. H. Johnson, Paul. R. Winn "Quantitative Methods for Management", 1976.
11. Perm Kumar Cupta, and Company Ltd. 1999 Ramngar, New Delhi 2005.
12. www.winterism.org.

PROGRAM (1)

SINGLE SERVER CASE

```
Type
   ar= array[0..100] of real;
VAR
   awt, avg , pis , avt, atcss  :real;
ac,pa ,pac,cs,ps,psc,ra,rs,tswq,tcss,idle:ar;
   x,y,s,a,sc,tse,tsb,atc,sima,sims:array[0..100] of integer;
   k,r,i,j,l,t,u,c,e,f,d,m,sums,suma,countidle:integer;
   sumsims,sumtswq,sumtcss,sumidlet:real;
procedure rand (var ar1:ar);
var
   v:integer;
begin
   Randomize;
   v:=1;
repeat
     ar1[v]:=Random(100)/100;
    inc(v);
until v>t;
END;
BEGIN
   AC[0]:=0;SC[0]:=0;PSC[0]:=0;x[0]:=0;Y[0]:=0;TSE[0]:=0;SIMA[0]:=0;
   sumsims:=0;sumtswq:=0;sumtcss:=0;sumidlet:=0;atc[0]:=0;
   writeln(' ENTER THE NUMBER OF TIME BETWEEN ARRIVAL:');
   read(k);
   writeln('ENTER OF SERVICE TIME);
   read( r);
   writeln('ENTER THE NUMBER OF SIMULATION:');
```

```
    read(t);
    for i:=1 to k do
begin
   writeln('enter arrival time');
   read(a[i]);
   ac[i]:=ac[i-1]+a[i];
   pa[i]:=1/k;
   pac[i]:=pac[i-1]+pa[i];
end;
   for j:=1 to r do
begin
   writeln('enter service time');
   read(s[j]);
]   sc[j]:=sc[j-1]+s[j];
   sums:=sums+s[j];
end;
   for j:=1 to r do
begin
   ps[j]:=s[j]/sums;
   psc[j]:=psc[j-1]+ps[j];
end;
   rand(ra);
   rand(rs);
   for l:= 1 to t do
begin
   for i:=1 to  k do
   if (ra[l]>pac[i])  and (ra[l]<=pac[i+1] )then sima[l]:=a[i];
   writeln(sima[l]);
end;
   for l:=1 to t do
begin
```

```
   for j:= 1 to r do
   if( rs[l]>psc[j]) and (rs[l]<=psc[j+1]) then sims[l]:=s[j];
end;
   for l:=1 to t do
begin
   if l=1 then atc[l]:=0
else
   atc[l]:=atc[l-1]+sima[l];
   if atc[l]>(sima[l-1]+sims[l-1]) then tsb[l]:= atc[l]
   else tsb[l]:=sima[l-1]+sims[l-1];
   tswq[l]:=atc[l]-tsb[l];
   tse[l]:=tsb[l]+sims[l];
   if tse[l]>atc[l] then tcss[l]:= tse[l]-atc[l]
else
   tcss[l]:=atc[l]-tse[l];
   if tsb[l]>tse[l-1] then idle[l]:=tsb[l]-tse[l-1]
else
   idle[l]:=tse[l]-tsb[l];
   sumsims:=sumsims+sims[l];
   sumtswq:=sumtswq+tswq[l];
   sumtcss:=sumtcss+tcss[l];
   sumidlet:=sumidlet+idle[l];
end;
   awt:=sumtswq/t;
   pis:=sumidlet/tse[t];
   avt:=sumsims/t;
   atcss:=sumtcss/t;
   for l:=1 to   t do
begin
   writeln('UNIT  NO:',l,' ','SIMA',sima[l],' ','SIMS',sims[l],' ','ATC',atc,'TSB' tsb[l]);
writeln('TSWQ','  ',tswq[l],' ','TSE',tse[l],' ','TCSS',tcss[l],' ','IDLE',idle[l]);
```

```
  writeln;
End;
  writeln( awt,' ', pis,' ',avt,' ',atcss);
  readln;readln;
end.
```

Where:

Sima: Simulating arrival time.

Sims: Simulating service time.

Tsb:Time service begin.

Tswq: Time units wait in the queue.

Tse: Time service end.

Tcss: Time units spend in system.

Idle: Idle time .

Awt: Average waiting time.

Pis: Prpbability of idle server.

Act: Average service time.

ATCSS: Average time units spend in system.

PROGRAM(2)

MULTIPLE SERVER CASE

```
TYPE
ar=array[0..100]of integer;
   arr=array[0..10,0..10]of integer;
   arra=array[0..100]of real;
   ara=array[0..10,0..10]of real;
VAR
   i,j,u,l,c,t,d,b,sumtiq :integer;
   cta,a,m,mint,minc,sima,max,tiq:ar;
   st,tse,tsb,se,s,sc,min:arr;
   pa,pac,rars:arra;
ps,psc:ara;
procedure rand (var ar1:arra);
VAR
   v:integer;
BEGIN
   Randomize;
   v:=1;
repeat
   ar1[v]:=Random(100)/100;
   inc(v);
until   v>t;
END;
BEGIN
   minc[C+1]:=1;PS[I,0]:=0;PSC[I,0]:=0;PA[0]:=0;PAC[0]:=0  ;min[i,0]:=0;
   SC[I,0]:=0;  AC[0]:=0;CTA[1]:=0;
   writeln('ENTER THE NUMBER OF THE SERVUCE CENTER');
   read(c )
   writeln('ENTER THE NUMBER OF ARRIVAL');
   read (r );
```

```
    wrireln('ENTER THE NUMBER OF SIMULATION');
 read(t );
     for i:=1 to c do
BEGIN
    writeln('ENTER THE  NUMBER OF SERVICE NUMBER');
read(m[i]);
END;
    for b:= 1 to r do
BEGIN
    writeln('ENTER ARRIVAL TIME');
    read(a[b]);
writeln ('INTER THE PROBABILITY OF INTER ARRIVAL TIME);
readln(pa[b]);
   pac[b]:=pac[b-1]+pa[b];
END;
Rand(a);
 For i:=1 to c do
BEGIN
      for b:= 1 to d do
BEGIN
if (ra[l]>pac[b]) and (ra[L]<=pac[b+1]) then
BEGIN
    sima[l]:=a[b];
    cta[L]:=cta[l-1]+sima[L];
END;
END;
END;
     for i:=1 to c do
     for j:=1 to m[i] do
BEGIN
    writeln('ENTER SERVICE TIME');
```

```
    read(s[i,j]);
    sc[i,j]:=sc[i,j-1]+s[i,j];
   writeln('ENTER THE PROBABILITY OF SERVICE TIME');
  readln(ps[i,j]);
psc[i,j]:=psc[i,j-1]+ps[i,j];
END;
     rand(rs);
    ps[i,0]:=0;
       for i:=1 to C do
        for l:=i to i do
BEGIN
   for j:=1 to m[l] do
BEGIN
    if  ((rs[l]>psc[l,j-1]) and (rs[L]<=psc[l,j])) then
BEGIN
   st[l,l]:=s[i,j];
   tsb[l,l]:=cta[l];
   tse[l,l]:=tsb[l,l]+st[l,l];
   min[l,l]:=tse[l,l];
END;
END;
END;
 min[i,0] :=0;
   for l:= c+1 to t do
BEGIN
     for u:=1 to l-1 do
   if min[minc[l],u]>min[minc[l],u-1] then mint[u]:=min[minc[l],u]
else
   mint[u]:=min[minc[l],u-1];

for i:=minc[l] to minc[l] do
```

```
BEGIN
   if ((rs[l]>psc[minc[l],j-1]) and (rs[l]<=psc[minc[l], j]))then
BEGIN
   st[minc[l],l]:=s[minc[l],j];
   if cta[l]> mint[u]then tsb[minc[l],l]:=cta[l]
   else    tsb[minc[l],l]:=mint[u];
   tse[minc[l],l]:=tsb[minc[l],l]+st[minc[l],l];
   min[minc[l],l]:=tse[minc[l],l];
  END;
END;
   if tse[minc[l],l]<=cta[l+1] then minc[l+1]:=minc[l]
else
begin
   if minc[l]=c then    minc[l+1]:=1
else
   minc[l+1]:= minc [l]+1 ;
END;
 END;
       for i:=1  to   c   do
    for l:=1  to   t    do
BEGIN
 If tse[I,l]>0 then tiq[l]:=tsb [i,l]-cta[l];
  writeln('TIQ:',tiq);
END;
    for i: 1 to t do
   sumtiq:=tiq[l-1]+tiq[l];
for i := 1 to c do
 st[t+1,i]:=0;
 for I := 1 to c  do
BEGIN
 for i : 1 to t   do
```

```
st[t+1,i]:=st[t+1,i]+st[i,l];
END;
    for i := 1 to c  do
    for  l: 1 to   t    do
BEGIN
        if (tsb[I,l] =0) and (st[i,l]=0) and (tse[i,l]=0) then
         write(‘ ‘,’ ‘,’ ‘);
       else
BEGIN
   writeln (‘units ‘,l,’ ‘,’ENTER SERVER’,I,’ ‘);
   write(‘TSB’,tsb[i,l],’ ‘);
   write(‘ST’,st[i,l],’ ‘);
   write(‘TSE’,tse[i,l],’ ‘);
  writeln;
END;
END;
     for i :1 to c do
 BEGIN
      ast[i]:=st[t+1]/t;
      writeln(‘AVERAGE SERVICE TIME FOR SERVER’,i,’=’,ast[i]);
END;
Atiq:=sumtiq/t;
Writeln(‘ATIQ’,atiq);
   readln;readln;
END.
```

Where:

Tse: Time service begin.

St: Service time.

Tse: Time service end.

Ast: Average service time.

Atiq:Average time in the queue.

PROGRAM(3)

POISSON DISTRIBUTION

```
Type
        ar=array[1..100]of integer;
        arr=array[1..100]of real;
VAR
        r,i,sum,fact:integer;
        a:ar;                   avg,pow :real;      pa:arr;
        function factorial(n:integer):integer;
BEGIN
        if n=1 then factorial:=1
        else  factorial:=n*factorial(n-1);
END;
        function power(m:real;s:integer):real;
VAR
        j:integer;              l:real;
BEGIN
        j:=1; power:=1;
         repeat
             l:=l*m;
             power:=l;
             j:=j+1;
        until  j>s;
END;
BEGIN                   Sum:=0;      pac[0]:=0;
       writeln('enter the number of arrival);
       readln( r) ;
       for i:=1 to r do
BEGIN
       writeln('enter Interarrival');
       read(a[i]);
```

```
        sum:=sum+a[i];
END;
        avg:=sum/r;
        for i:=1 to r do
BEGIN
       fact:=factorial(i );
       pow:=power(avg,i);
       pa[i]:=(exp(-avg)*pow)/fact;
       pac[i]:=pac[i-1]+pac[i];
END;
        readln; readln;
END.
```

PROGRAM(4)

EXPONENTIAL DISTRIBUTION

```
VAR
    i,k,sum:integer;
    avg:real;
    a:array[1..100]of integer;
    ps,psc:array[0..100]of real;
BEGIN                   psc[0]:=0;
 ;('writeln('ENTER :K
       readln(k);
      for i:=1 to k  do
BEGIN
      writeln('ENTER  TIME BETWEEN ARRIVAL );
     read(a[i]);
    sum:=sum+a[i];
END;
    avg:=sum/k;
    writeln(avg);
    for i:=1 to k  do
BEGIN
    ps[i]:=avg*(exp(-avg*a[I]));
    psc[i]:=psc[i-1]+ps[i];
END;
    for i:= 1 to k  do
BEGIN
    writeln(ps[i]);
    writeln(psc[i]);
END;
    readln;readln;
.END
```

PROGRAME(5)

RANDOM NUMBER GENERATION

```
TYPE
      ar=array[0..100]of integer ;
      arr=array[0..100]of real;
VAR
    a,c,i,t:integer;
    x:ar;
    r:arr;
BEGIN
    writeln('ENTER A:B:X[O]');
    readln(a);
     readln(c);
    readln(x[0]);
    writeln('ENTER THE NUMBER OF SIMULATION');
    readln(t );
    for i:=1 to t  do
BEGIN
    x[i]:=(a*x[i-1]+c)  mod 100;
    r[i]:=x[i]/100;
END;
    for i:=1 to  t  do
    writeln(r[i]);
    readln;readln;
END.
```