Simulation Method:

```
function[]=valueofbackups(vA,vB,vC,fp,n,k)

%Input Arguments:
%vA=[0 .01 .04 .08 .14 .21 .29 .39 .5 .61 .71 .79 .86 .92 .96 .99 1]
%vB=[0 0 0 0 0 .1 .3 .6 1 1 1 1 1 1 1 1 1 ]
%vC=[0 0 .03 .08 .14 .21 .29 .38 .48 .58 .67 .75 .82 .88 .93 .96 1]
%fp=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16];
%n= n-by-n matrix of uniform samples
%k= k best players

%input vector vA for player A probabilities row vector
%input vector fA for player A fantasy point row vector
%input n for an N-by-N matrix of uniform random sample

[vArow,vAcol]=size(vA);
[fprow,fpcol]=size(fp);

W1=zeros(n); %Recording the Fantasy point according to the simulation
W2=zeros(n);
W3=zeros(n);
Z1=rand(n);  %Generating N-by-N matrix of uniform random sample
Z2=rand(n);
Z3=rand(n);

sumofbestk=zeros(n); %Record sum of best k players
fpscount=zeros(2*(vAcol-1),1); %count occurance of FPs from 1 to k*16

for ii=1:n
    for jj=1:n
        for kk=1:vAcol-1
            if vA(kk)<Z1(ii,jj) && Z1(ii,jj)<=vA(kk+1);
                W1(ii,jj)=fp(kk+1);
            end
            if vB(kk)<Z2(ii,jj) && Z2(ii,jj)<=vB(kk+1);
                W2(ii,jj)=fp(kk+1);
            end
            if vC(kk)<Z3(ii,jj) && Z3(ii,jj)<=vC(kk+1);
                W3(ii,jj)=fp(kk+1);
            end
        end
        s=sort([W1(ii,jj) W2(ii,jj) W3(ii,jj)],'descend');
        sumofbestk(ii,jj)=sum(s(1:k));
        for kkk=1:2*(vAcol-1)
            if kkk-1<sumofbestk(ii,jj) && sumofbestk(ii,jj)<=kkk
                fpscount(kkk)=fpscount(kkk)+1;
            end
        end
    end
end
fps=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32]';
result=[fps fpscount];

W1;
W2;
W3;
Z1;
Z2;
Z3;
sumofbestk;
```

```
result
```
testruns:
W1 =

```
    9     4    11
    6    10    13
   10    10    15
```

W2 =

```
    8     7     8
    7     5     7
    8     6     8
```

W3 =

```
    5     3     3
    9    10     9
   10     7     5
```

Z1 =

```
   0.5860    0.0835    0.7298
   0.2467    0.6260    0.8908
   0.6664    0.6609    0.9823
```

Z2 =

```
   0.7690    0.5801    0.8627
   0.5814    0.0170    0.4843
   0.9283    0.1209    0.8449
```

Z3 =

```
   0.2094    0.0320    0.0495
   0.5523    0.6147    0.4896
   0.6299    0.3624    0.1925
```

sumofbestk =

```
   17    11    19
   16    20    22
```

```
   20   17   23
result =

    1    0
    2    0
    3    0
    4    0
    5    0
    6    0
    7    0
    8    0
    9    0
   10    0
   11    1
   12    0
   13    0
   14    0
   15    0
   16    1
   17    2
   18    0
   19    1
   20    2
   21    0
   22    1
   23    1
   24    0
   25    0
   26    0
   27    0
   28    0
   29    0
   30    0
   31    0
   32    0
```

```
>> valueofbackups(vA,vB,vC,fp,4,2)

W1 =

    4    3   10    4
    5   10    8    8
    5    6    9   12
    5    9    8   13


W2 =

    6    7    6    7
    6    6    6    5
    7    8    6    8
    8    5    8    8


W3 =

   11   16    5   15
    3   13    8    7
   13   12    4    7
   15    9    3    7


Z1 =

   0.1231   0.0427   0.6952   0.1239
   0.2055   0.6352   0.4991   0.4904
   0.1465   0.2819   0.5358   0.8530
   0.1891   0.5386   0.4452   0.8739


Z2 =

   0.2703   0.4170   0.1057   0.5737
   0.2085   0.2060   0.1420   0.0521
   0.5650   0.9479   0.1665   0.9312
   0.6403   0.0821   0.6210   0.7287


Z3 =

   0.7378   0.9844   0.1776   0.9391
   0.0634   0.8589   0.3986   0.3013
   0.8604   0.7856   0.1339   0.2955
   0.9344   0.5134   0.0309   0.3329
```

sumofbestk =

17	23	16	22
11	23	16	15
20	20	15	20
23	18	16	21

result =

1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	1
12	0
13	0
14	0
15	2
16	3
17	1
18	1
19	0
20	3
21	1
22	1
23	3
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0

Exact Method

```
function[]=valuebackup(A,B,C)
A=[0.01 0.03 0.04 0.06 0.07 0.08 0.1 0.11 0.11 0.1 0.08 0.07 0.06 0.04 0.03
0.01];
B=[0 0 0 0 .1 .2 .3 .4 0 0 0 0 0 0 0 0];
C=[0 .03 .05 .06 .07 .08 .09 .1 .1 .09 .08 .07 .06 .05 .03 .04];

V=zeros(1,32);
F=zeros(1,32);

for ii=1:32
    F(ii)=ii;
end

for ii=1:16
    for jj=1:16
        for kk=1:16
            p=A(ii)*B(jj)*C(kk);
            S=sort([ii,jj,kk],'descend');
            sumofbest2=sum(S(1:2));
            V(sumofbest2)=V(sumofbest2)+p;
        end
    end
end

V=[F' V'];

V
```

V =

1.000000000000000	0
2.000000000000000	0
3.000000000000000	0
4.000000000000000	0
5.000000000000000	0
6.000000000000000	0
7.000000000000000	0.000120000000000
8.000000000000000	0.000760000000000
9.000000000000000	0.002720000000000
10.000000000000000	0.007130000000000
11.000000000000000	0.014300000000000
12.000000000000000	0.025260000000000
13.000000000000000	0.038750000000000
14.000000000000000	0.056200000000000
15.000000000000000	0.073700000000000
16.000000000000000	0.089430000000000
17.000000000000000	0.095150000000000
18.000000000000000	0.094940000000000
19.000000000000000	0.091310000000000
20.000000000000000	0.086540000000000
21.000000000000000	0.078980000000000
22.000000000000000	0.067620000000000
23.000000000000000	0.054930000000000

24.000000000000000	0.042560000000000
25.000000000000000	0.026700000000000
26.000000000000000	0.019600000000000
27.000000000000000	0.013600000000000
28.000000000000000	0.009100000000000
29.000000000000000	0.005700000000000
30.000000000000000	0.003000000000000
31.000000000000000	0.001500000000000
32.000000000000000	0.000400000000000