

#include<algorithm>

#include<cstdio>

#define mxn 110

using namespace std;

int n,m,K,sl,ans,w[mxn],s[mxn],d[mxn],siz[mxn],pre[mxn],sum[mxn],val[mxn],dp[mxn][510];

int tim,top,id[mxn],stk[mxn],vis[mxn],dfn[mxn],low[mxn];

int t,h[mxn];

struct edge

{

int fr,to,nxt;

}e[mxn];

inline char gc()

{

static char buf[1<<16],\*S,\*T;

if(S==T){T=(S=buf)+fread(buf,1,1<<16,stdin);if(S==T)return EOF;}

return \*S++;

}

inline int rd()

{

sl=0;

char ch=gc();

while(ch<'0'||'9'<ch) ch=gc();

while('0'<=ch&&ch<='9') sl=sl\*10+ch-'0',ch=gc();

return sl;

}

inline void add(int u,int v) {e[++t]=(edge){u,v,h[u]};h[u]=t;}

void tarjan(int u)

{

dfn[u]=low[u]=++tim;

stk[++top]=u;

vis[u]=1;

int v;

for(int i=h[u];i;i=e[i].nxt)

if(!dfn[v=e[i].to])

tarjan(v),low[u]=min(low[u],low[v]);

else if(vis[v]) low[u]=min(low[u],dfn[v]);

if(dfn[u]==low[u])

{

++K;

do

{

v=stk[top--];

sum[K]+=w[v];

val[K]+=s[v];

vis[v]=0;

id[v]=K;

}while(u!=v);

}

}

void dfs(int u)

{

int v;dfn[++tim]=u;siz[u]=1;

for(int i=h[u];i;i=e[i].nxt)

pre[v=e[i].to]=pre[u]+sum[u],

dfs(v),siz[u]+=siz[v];

}

inline void upd(int &x,int y) {if(y>x) x=y;}

int main()

{

n=rd();m=rd();int x;

for(int i=1;i<=n;++i) w[i]=rd();

for(int i=1;i<=n;++i) s[i]=rd();

for(int i=1;i<=n;++i)

{

x=rd();

if(x) add(x,i);

}

for(int i=1;i<=n;++i)

if(!dfn[i])

tarjan(i);

top=t;t=0;fill(h+1,h+K+1,0);

for(int i=1;i<=top;++i)

if(id[e[i].fr]!=id[e[i].to])

d[id[e[i].to]]++,add(id[e[i].fr],id[e[i].to]);

for(int i=1;i<=K;++i)

if(!d[i])

add(0,i);

tim=0;dfs(0);

for(int i=1;i<=tim;++i)

{

for(int j=pre[dfn[i]];j<=m-sum[dfn[i]];++j)

upd(dp[i+1][j+sum[dfn[i]]],dp[i][j]+val[dfn[i]]);

for(int j=pre[dfn[i]];j<=m;++j)

upd(dp[i+siz[dfn[i]]][j],dp[i][j]);

}

printf("%d\n",dp[tim+1][m]);

return 0;

}

