Castle in C++  
  
Please help me to convert this to C#

1. #include<iostream>
2. #include<vector>
3. #include<string.h>
4. #include<string>
5. #include<stdio.h>
6. #include<stdlib.h>
7. #include<algorithm>
8. **using** **namespace** std;
9. **const** **int** MaxN=109;
10. **const** **int** INF=1000000000;
11. **int** N,M;
12. vector<**int**> e[MaxN];
13. vector<pair<**int**,**int**>> Go[MaxN];
14. **int** A[MaxN],B[MaxN],F[MaxN],G[MaxN];
16. **void** init()
17. {
18. **for**(**int** i=1;i<=N;i++)
19. e[i].clear();
20. **for**(**int** i=1;i<=N;i++)
21. {
22. **int** x;
23. scanf("%d%d%d",&A[i],&B[i],&x);
24. B[i]+=x;
25. **if**(A[i]>=B[i]);
26. **else**
27. A[i]=B[i];
28. }
29. **for**(**int** i=1;i<N;i++)
30. {
31. **int** u,v;
32. scanf("%d%d",&u,&v);
33. e[u].push\_back(v);
34. e[v].push\_back(u);
35. }
36. }
38. **void** Dfs(**int** fa,**int** x)
39. {
40. F[x]=A[x];
41. G[x]=B[x];
42. Go[x].clear();
43. **for**(**int** i=0;i<e[x].size();i++)
44. **if**(e[x][i]!=fa)
45. {
46. Dfs(x,e[x][i]);
47. Go[x].push\_back(make\_pair(G[e[x][i]],F[e[x][i]]));
48. }
49. sort(Go[x].begin(),Go[x].end());
50. **for**(**int** i=Go[x].size()-1;i>=0;i--)
51. {
52. **if**(F[x]-G[x]<Go[x][i].second)
53. F[x]=G[x]+Go[x][i].second;
54. G[x]+=Go[x][i].first;
55. }
56. }
58. **void** solve()
59. {
60. **int** ans=INF;
61. **for**(**int** i=1;i<=N;i++)
62. {
63. **for**(**int** j=1;j<=N;j++)
64. F[j]=G[j]=-1;
65. Dfs(0,i);
66. **if**(F[i]<ans)
67. ans=F[i];
68. }
69. printf("%d\n",ans);
70. }
72. **int** main()
73. {
74. **int** T=0;
75. **while**(1==scanf("%d",&N)&&N!=0)
76. {
77. T++;
78. init();
79. printf("Case %d: ",T);
80. solve();
81. }
82. **return** 0;
83. }

#include<iostream>

#include<vector>

#include<string.h>

#include<string>

#include<stdio.h>

#include<stdlib.h>

#include<algorithm>

using namespace std;

const int MaxN=109;

const int INF=1000000000;

int N,M;

vector<int> e[MaxN];

vector<pair<int,int>> Go[MaxN];

int A[MaxN],B[MaxN],F[MaxN],G[MaxN];

void init()

{

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

e[i].clear();

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

{

int x;

scanf("%d%d%d",&A[i],&B[i],&x);

B[i]+=x;

if(A[i]>=B[i]);

else

A[i]=B[i];

}

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

{

int u,v;

scanf("%d%d",&u,&v);

e[u].push\_back(v);

e[v].push\_back(u);

}

}

void Dfs(int fa,int x)

{

F[x]=A[x];

G[x]=B[x];

Go[x].clear();

for(int i=0;i<e[x].size();i++)

if(e[x][i]!=fa)

{

Dfs(x,e[x][i]);

Go[x].push\_back(make\_pair(G[e[x][i]],F[e[x][i]]));

}

sort(Go[x].begin(),Go[x].end());

for(int i=Go[x].size()-1;i>=0;i--)

{

if(F[x]-G[x]<Go[x][i].second)

F[x]=G[x]+Go[x][i].second;

G[x]+=Go[x][i].first;

}

}

void solve()

{

int ans=INF;

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

{

for(int j=1;j<=N;j++)

F[j]=G[j]=-1;

Dfs(0,i);

if(F[i]<ans)

ans=F[i];

}

printf("%d\n",ans);

}

int main()

{

int T=0;

while(1==scanf("%d",&N)&&N!=0)

{

T++;

init();

printf("Case %d: ",T);

solve();

}

return 0;

}