```cpp
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <algorithm>
#include <vector>

using namespace std;

vector<vector<int> > lnklst;
vector<bool> visited ;
vector<int> result;

void dfs(int r) {
  if (visited [r]) return;
  for (int i=0; i<lnklst[r].size(); i++) {
     dfs(lnklst[r][i]);
  }
  visited [r] = true;
  result.push_back(r);
  return;
}

vector<int> topsort(const vector<vector<int> > &lnklst) {
  int n = lnklst.size() - 1;
  visited .clear();
  visited .resize(n+1);
  result.clear();
  for (int i=n; i>0; i--) {
     if (!visited [i]) dfs(i);
  }
  reverse(result.begin(), result.end());
  return result;
}

int main() {
  FILE *fin;

  /*fin = fopen("topsort.dat", "r");
  assert(fin);*/
  fin = stdin;
  int n, m;
  fscanf(fin, "%d %d\n", &n, &m);
  lnklst.resize(n+1);
  for (int i=0; i<m; i++) {
     int u, v;
     fscanf(fin, "%d %d\n", &u, &v);
     lnklst[u].push_back(v);
  }
  vector<int> result = topsort(lnklst);
  for (int i=0; i<result.size(); i++) {
     printf("v%d ", result[i]);
  }
  printf("\n");
  return 0;
}
```