```
#include <iostream>
#include <vector>
//// MISSING LINE HERE 1
class Node;
void spanNodes(Node * , int,string );
int main();
class Link
{
public:
        int cost,
        Node*dest;
        //// MISSING LINE HERE 2
};
class Node
{
        //// MISSING LINE HERE 3
public:
        char name;
        //// MISSING LINE HERE 4
        vector<Link*> neighbors;
        static void addNode(char name) { Node* node = new Node(name); }
        static bool addJoint(Node * p1 , Node* p2, int cost);
                p1.neighbors.push_back(new Link(cost,p2));
                return true;
        }
private:
        static vector<Node*> nodes;
};
vector<Node*> Node::nodes = vector<Node*>();
void spanNodes(Node * startPoint , int budget, string path)
{
        if (path == "") path = startPoint->name;
        for (int j = 0; j < startPoint->neighbors.size(); j++) {
                if (startPoint->neighbors[j]->cost < budget)
                {
                        cout << path << startPoint->neighbors[j]->dest->name << endl;
                        spanNodes(startPoint->neighbors[j]->dest,
                                budget - startPoint->neighbors[j]->cost,
                                path + startPoint->neighbors[j]->dest->name);

                }

        }
}
int main()
{
        Node::addNode('A');
        Node::addNode('B');
        Node::addNode('C');
        Node::addNode('D');
        Node::addJoint(Node::nodes[0], Node::nodes[1], 4);
        Node::addJoint(Node::nodes[3], Node::nodes[2], 3);
        Node::addJoint(Node::nodes[0], Node::nodes[3], 3);
        Node::addJoint(Node::nodes[1], Node::nodes[3], 3);
        Node::addJoint(Node::nodes[3], Node::nodes[0], 3);
        spanNodes(Node::nodes[0], 14,"")
}
```

OUTPUT:
AB
ABC
ABD
ABDA
ABDAB
ABDAC
AC

4.

a) (10p) Write four missing lines without touching other codes to eliminate any compile errors.

b) (20p) Extend spanNodes function to write the resulting paths in a global array or vector variable that can be populate any time without rerunning spanNodes function.

c) (20p) Re write addJoint function that can take names of Nodes instead of its address information. Prototype: **static bool addJoint(char source, char dest , int cost);**

d) (20p) addJoint function has a Boolean return type to show if the link is acceptable, however the programmer misses to implement the checks for wrong inputs to the function and duplicate link entries. Correct the function and write these control codes. Return false for abnormal cases.

You Have 160 points question try at least 100. You can solve as much as you can. Best of luck