Morgan Baccus
CptS 317
Homework 10

1.

```
#include <bits/stdc++.h>
using namespace std;

bool go (bool a, bool b, bool c, bool d)
{
int z = pow(2,10);
if (z == 1024)
return true;
else
return false;
}

bool p (bool a, bool b, bool c, bool d, int n)
{
bool y = go(a, b, c, d);
if (n==1)
return true;

bool r = go (a, b, c, d);
n = n-1;
bool v = p(a, b, c, r, n);
return v;
}

int main()
{
bool a, b, c, d;
a = false, b = false;
c = true, d = true;
bool x = p(a, b, c, d, 5);
cout<<x<<”\n”;
}
```

2.

Consider the given graph G.

For the graph G, we can convert it as transition diagram of finite state automata with vertices represented as states and edges are treated as transition. Then the alphabet of this transition is equal to colors of the vertices in the graph.

Now the label of transition from state p to q will be equal to color of the outgoing vertex I.e. color of vertex p.

In this way we will convert given graph into finite automata with start state equal to start node and accepting state will be equal to end node of the graph.

Now once the equivalent finite state automata is obtained from the given graph G, the regular expression of this given finite automata will be equal to list of all possible walks from start state to accepting state and this regular expression corresponding to the finite automata can be obtained algorithmically.

Once the regular expression is obtained, add the alphabet corresponding to color of the end node because that was yet to be added into the regular expression. Now we have to check whether the regular expression contains any possible string with (1) alphabet corresponding to number of red color is greater than alphabet corresponding to number of green color and (2) number of alphabet corresponding to blue color is at least 5 and number of alphabet corresponding to yellow color is at most 3.

If both conditions are satisfied by the regular expression (this is decidable problem) then accept, else reject.

Thus, we have decidable solution to above problem and hence this problem is decidable.

To summarize the steps:

1. For the given graph G, convert it as transition diagram of finite state automata, with vertices as states, edge as transition function, color of vertices as alphabet, color of source vertex in an edge as the input alphabet of the transition function. Finally start node is the start state, end node as the only accepting state.

2. Convert the transition diagram of finite state automata into regular expression.

3. Check whether any string exist in the regular expression which satisfies the criteria given in question.

4. Thus, we get the decidable solution.