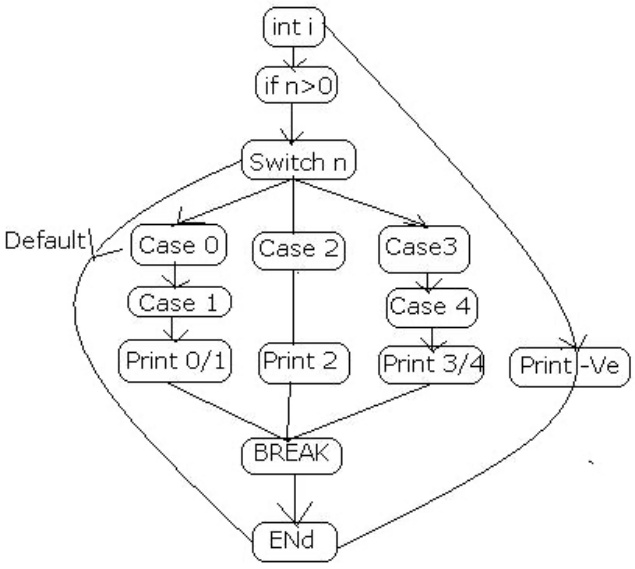
Analysis of the code:

```
if (n >= 0) {
    switch(n) {
        case 0:
        case 1:
            printf("zero or one\n");
            break;
        case 2:
            printf("two\n");
            break;
        case 3:
        case 4:
            printf("three or four\n");
            break;
        }
    }
else {
    printf ("negative\n");
}
```

Result Obtained:



Edges 16 , Nodes 14
V(G) -> (16-14)+ 2--> 5

Code Chef:

Optimising complexity code

```
private static void replaceNodeProperties(HashMap
configNodeProperties, JSONArray esdNodepropertiesList)
{

    HashMap replaceConfigNodePropRules = (HashMap)
configNodeProperties.get("replace");

    if (replaceConfigNodePropRules != null) {
      for (Object replaceConfigNodeProp :
replaceConfigNodePropRules.keySet()) {
        for (int j = 0; j < esdNodepropertiesList.size(); j+
+) {
          JSONObject esdNodePropObj = (JSONObject)
esdNodepropertiesList.get(j);
          if
(esdNodePropObj.get("name").toString().contains("::")) {
            esdNodePropObj.put("name",
esdNodePropObj.get("name").toString().split("::")[1]);
          }
          if
(esdNodePropObj.get("name").equals(replaceConfigNodeProp)) {
            if (replaceConfigNodeProp.equals("resourceType"))
{
              esdNodepropertiesList.remove(j);
              ArrayList replaceNodePropList = (ArrayList)
replaceConfigNodePropRules.get(replaceConfigNodeProp);

esdNodepropertiesList.addAll(replaceNodePropList);
            } else if
(replaceConfigNodeProp.equals("comments"))
              esdNodePropObj.put("name",
replaceConfigNodePropRules.get(replaceConfigNodeProp));
            break;
          }
        }
      }
    }
  }
```