```
bool parseModelNumber(const string& modelnumber, string& model,  string& port, bool&
 stacking) {
     stringstream ss(modelnumber);
     string part;
     vector< string> parts;

    while ( getline(ss, part, '-')) {
        parts.push_back(part);
    }

    if (parts.size() < 2 || !isdigit(parts[1][0])) {
        return false;
    }

    model = "";
    for (char c : parts[0]) {
        if (isdigit(c)) {
            model += c;
        }
    }
    port = "";
    for (char c : parts[1]) {
        if (isdigit(c)) {
            port += c;
        }
    }
    stacking = (modelnumber.find("S") !=  string::npos) || (modelnumber.find("NX") !
=  string::npos);

    return true;
}

void classifySwitch(const  string& modelnumber) {
     string model, port;
    bool stacking;

    if (!parseModelNumber(modelnumber, model, port, stacking)) {
         cout << "Invalid model number" <<  endl;
        return;
    }

    if ((model == "5200" || model == "5250" || model == "5270") && stoi(port) <= 24)
 {
         cout << port <<  endl;
         cout << "Type 1" <<  endl;
    } else if ((model == "5200" || model == "5250" || model == "5270" || model == "5
300" || model == "5350" || model == "5370") && stoi(port) > 24 && model.find("5400")
 ==  string::npos) {
         cout << port <<  endl;
         cout << "Type 2" <<  endl;
    } else if (stacking) {
         cout << port <<  endl;
         cout << "Core" <<  endl;
    } else {
         cout << "Invalid model number" <<  endl;
    }
}
```