

Discover

Discover card — multiple BIN ranges.

```
^6(011|5\d{2}|4[4-9]\d|22[1-9]\d{2})\d{12}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
6	literal text "6"
(	start of a capturing group
011	literal text "011"
	OR — try the alternative
5	literal text "5"
\d{2}	exactly 2 times: digit (0–9)
	OR — try the alternative
4	literal text "4"
[4-9]	any of: 4–9
\d	any digit (0–9)
	OR — try the alternative
22	literal text "22"
[1-9]	any of: 1–9
\d{2}	exactly 2 times: digit (0–9)
)	end of group
\d{12}	exactly 12 times: digit (0–9)
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript	<pre>const re = /^6(011 5\d{2} 4[4-9]\d 22[1-9]\d{2})\d{12}\$/; re.test(value);</pre>
Python	<pre>import re re.match(r"^6(011 5\d{2} 4[4-9]\d 22[1-9]\d{2})\d{12}\$", value)</pre>
Java	<pre>Pattern.compile("^6(011 5\\d{2} 4[4-9]\\d 22[1-9]\\d{2})\\d{12}\$").matcher(value).match es();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"^6(011 5\d{2} 4[4-9]\d 22[1-9]\d{2})\d{12}\$");</pre>
Go	<pre>regexp.MustCompile(`^6(011 5\d{2} 4[4-9]\d 22[1-9]\d{2})\d{12}\$`).MatchString(value)</pre>
Ruby	<pre>/^6(011 5\d{2} 4[4-9]\d 22[1-9]\d{2})\d{12}\$/ .match?(value)</pre>
PHP	<pre>preg_match('~^6(011 5\d{2} 4[4-9]\d 22[1-9]\d{2})\d{12}\$~', \$value);</pre>

Common pitfalls

- **Anchored to the whole string.** This pattern uses `^` and `$`, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (`g`) flag.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (`\\d`); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the discover against a source of truth (database, API, or checksum) where it matters.