

Könnyű álmok (13. rész)

A titkosítás titokzatos titkai...

Ebben a hónapban egy kis kitérőt teszünk. Utolsó két cikkünk az Interneten leggyakrabban használt hálózati protokollok sajátosságait tárta fel. Ebbe az irányba haladva mindenképpen szót kell ejtenünk azokról a protokollokról és protokollkiegészítésekről, amelyek az átvitt adatokat valamilyen formában titkosítják. Az elkövetkező néhány cikk a titkosításról és annak gyakorlati alkalmazásairól szól. A téma elérhető szakirodalma kimerítő, sőt magyar nyelvű források is léteznek, ennek ellenére a felhasználók jelentős része úgy tekint a kriptográfiára, mint az űrtudományra – messziről. A titkosítás tudományának eredményeiben vakon bízunk, de a mögötte dolgozó algoritmusok és eljárások nagyfokú bonyolultsága visszariasztja a közelebbi ismeretségtől. Fontosnak tartjuk, hogy végre egy olyan összefoglaló szülessen, amely nem igényel komoly matematikai előképzettséget, de a fontosabb eljárások alapelveit mégis meg lehet belőle érteni. Tévedés ne essék, ez a pár oldal a tárgyalt téma pontos, mérhető adatokra alapozott leírása és bizonyítása híján csak tudományos ismeretterjesztő cikknek tekinthető. Reméljük azonban, hogy az itt leírtak sokakban kedvet ébresztenek a titkosítás tudományának behatóbb tanulmányozására. Az érdeklődők az ajánlott irodalomban minden kérdésére kimerítő választ találnak. Akik pedig nem válnak kriptográfussá, azok betekintést nyernek ebbe a komoly és érdekes tudományba. Helyezkedjenek el kényelmesen, és jó szórakozást!

A titkosításról általában

A titkosítás szép, hosszú története helyhiány miatt ma kimarad. Inkább foglalkozunk a gyakorlattal. Igyekszünk csak arról szólni, ami működésének megértéséhez elengedhetetlenül szükséges. Mi is a titkosítás? Az adatok titkosságával, hitelességével és védettségével foglalkozó tudomány. Két alapvető ága van: a kriptográfia, amely a korábban felvetett kérdések megoldásával foglalkozik; és a kriptanalízis, amelynek a célja a kriptográfia által adott megoldások hibáinak feltárása. A kriptográfia további két fontosabb részre osztható. Az egyik ága a titkosító algoritmusokkal foglalkozik, a másik azok használati körülményeivel, ahol ezen eljárásokat összefoglaló néven kriptográfiai protokolloknak hívják.

A kriptográfia alapfogalmai

A kriptográfiának, mint minden tudománynak, megvannak a saját szakkifejezései. Ezeket a későbbiek egyértelműsége és a vonatkozó szakirodalom értő tanulmányozásához előnyös megismerni. Az adatok továbbítása általában két fél között történik: a küldő (sender) és a fogadó (receiver) között. Az átvivendő szöveg eredeti formájában a nyílt szöveg (plaintext), titkosítva pedig kriptoszöveg (ciphertext). A nyílt szöveget kriptoszöveggé alakító eljárás a titkosítás (encryption), ellentéte pedig a visszafejtés (decryption). A korszerű titkosító eljárások egy vagy több úgynevezett kulcsot (key) használnak. A kulcs egy, az algoritmus által megkövetelt módon előállított, általában kis méretű adatdarab. A jó



titkosító

algoritmusok alapvető

tulajdonsága, hogy a kulcs ismerete nélkül a nyílt szöveget a kriptoszövegből még az algoritmus ismeretében sem lehet visszanyerni.

A jobb érthetőség végett szerepeljenek a kapcsolattartás lépései a fenti kifejezések használatával:

- A *küldő* egy üzenetet megbízhatatlan (lehallgatható) csatornán szeretne a *fogadónak* továbbítani.
- A *nyílt szöveget* valamilyen eljárással, a *kulcs* segítségével *titkosítja*, így előáll a *kriptoszöveg*.
- A *kriptoszöveget* már nyugodtan elküldheti a megbízhatatlan csatornán a *fogadónak*, mivel azt – reményeink szerint – más nem tudja elolvasni.
- A *fogadó* a kapott *kriptoszöveget* a *kulcs* segítségével *visszafejti* (amely nem feltétlenül egyezik meg a titkosító kulccsal), így a *nyílt szöveget* már elolvashatja.

A titkosító eljárások kezdetben semmilyen kulcsokat nem igényeltek, de a népek hamar rájöttek, hogy az ilyen eljárások megbízhatósága nem kielégítő, ráadásul megfejtésük esetén nehéz őket lecserelni. A későbbiekben olyan eljárások kidolgozására törekedtek, ahol a titkosítás megfejtéséhez egy titkos szó vagy könyv, illetve valamilyen eszköz birtoklására is szükség volt. Az ilyen eljárások nagy előnye, hogy ha nem az algoritmus hibás, hanem csak a titkos „kulcs” vált ismertté, annak cseréjével a titkosság helyreállt. A haladó titkosító eljárások is használnak ilyen kulcsokat, de itt a kulcs már valamilyen előre meghatározott tulajdonságokkal bíró adathalmaz, amely elengedhetetlen az algoritmus végrehajtásához. A korábbi kulcsalapú módszerek mai megfelelői, a szimmetrikus titkosítási eljárások egyetlen titkos kulcsot használnak. E módszerek komoly hátránya, hogy a küldőnek és a fogadónak előre meg kell állapodnia a kulcsban. Itt a kulcs biztos továbbítása jelenti a gondot. Ennek kiküszöbölésére hozták létre az aszimmetrikus algoritmusokat, amelyek már nem igénylik a titkos kulcscserét. Mindkét félnek van egy titkos és egy nyilvános kulcsa, és ezen kulcspár segítségével meg tudják oldani a kulcscserét, adott esetben magát az adatátvitelt is. A fent leírtak kissé „matematikusabb” írásmóddal így néznek ki:

Szimmetrikus eljárások (egyetlen kulccsal)

- $E(p,k) = c$
- $D(c,k) = p$

tehát

- $D(E(p,k),k) = p$

ahol

- E – titkosítás (encrypt)
- D – visszafejtés (decrypt)
- p – nyílt szöveg (plaintext)

Eltolási táblázat az angol ábécére:

```
abcdefghijklmnopqrstuvwxyz
efghijklmnopqrstuvwxyzabcd
```

A nyílt szöveg:

Kínos, ha az embert fejbe vágják egy lédús citromszelettel, amit egy aranytáglára erősítettek.

A kriptoszöveg:

omrsw, le ed iqfivx jinfi zekneo ikc pihyw gmxxsqwdipixxi, eqmx ikc evercxikpeve ivswmxixxio.

1. ábra A Caesar-féle titkosítás működése

- c – kriptoszöveg (ciphertext)
- k – kulcs (key)

Aszimmetrikus algoritmusok (több kulccsal)

- $E(p, k_1) = c$
- $D(c, k_2) = p$

tehát

- $D(E(p, k_2), k_1) = p$

A lenyomatokról (hash)

És most valami egészen más. A titkosítás kihívásainak megoldása közben olyan eljárások kidolgozása is szükségessé vált, amelyek egy adathalmazról (akár egy nyílt szövegről) képesek lenyomatot (hash) képezni. Ezt az eljárást egyirányú függvénynek vagy kriptográfiai ellenőrzőösszeg-képzésnek, magát a lenyomatot pedig más néven kivonatnak, ujjlenyomatnak vagy kriptográfiai ellenőrző összegnek is hívják. Mire jó ez a gyakorlatban? Tételezzük fel, hogy van egy olyan eljárásunk, amelynek segítségével egy tetszőlegesen nagy méretű adatot át tudunk alakítani egy olyan lényegesen kisebb kötött méretűvé, amelyre igazak az alábbi állítások:

- A lenyomattól az eredeti adata vonatkozóan semmilyen következtetést nem lehet levonni, vagyis *egyirányú*.
- Nagyon nehezen található olyan másik adat, amelynek lenyomata megegyezik az eredeti adat lenyomatával, tehát nagymértékben *ütközésmentes*.
- Az eredeti adat kismértékű változása nagy változást okoz a lenyomatban. Ezt a szakirodalom *lavinahatásnak* hívja.

Ha ilyen eljárással képzünk lenyomatot a nyílt szövegről, a lenyomat titkosított változatát az átvitt szöveg mellé csatolva a másik oldal meggyőződhet róla, hogy az üzenetet valóban mi küldtük, és az átvitel során az nem módosult. Sokakban felmerülhet a kérdés, hogy miért nem a teljes átvendő adatot titkosítottuk. Ennek két oka van: az egyik, hogy bizonyos esetekben az üzenet titkosítására nincs szükség, csak feladójának egyértelmű azonosítása a cél. A másik ok, hogy a titkosító algoritmusok jelentős része nem gyors, sőt kifejezetten lassú. Egy mai átlagos gépen a később tárgyalt RSA algoritmus alkalmazása egy 10 MB méretű állományra néhány perc.

A jó lenyomatképző algoritmusok fontosabb elvárt tulajdonságait fent már felsoroltuk. Tisztázzuk azonban, hogy pontosabban mit jelent az ütközésmentes kifejezés. Mivel

A betűhelyettesítő táblázat

```
abcdefghijklmnopqrstuvwxyz
bcdefghijklmnopqrstuvwxyzabcd
cdefghijklmnopqrstuvwxyzabcde
efghijklmnopqrstuvwxyzabcdef
ghijklmnopqrstuvwxyzabcdefg
hijklmnopqrstuvwxyzabcdefgh
ijklmnopqrstuvwxyzabcdefghi
jklmnopqrstuvwxyzabcdefghijk
klmnopqrstuvwxyzabcdefghijklm
lmnopqrstuvwxyzabcdefghijklmn
mnopqrstuvwxyzabcdefghijklmno
pqrstuvwxyzabcdefghijklmnopqr
qrstuvwxyzabcdefghijklmnopqrst
vwxyzabcdefghijklmnopqrstuvw
xyzabcdefghijklmnopqrstuvwxy
zabcdefghijklmnopqrstuvwxy
```

A nyílt szöveg:

Frodó úr szerzett egy gyűrűt, és mindenhová magával hurcolta, ha kellett, ha nem.

A titkosító jelszó:

cirok se prucirok sep ruciro ks epruciroks eprucir oksepruc ir oksepru ci rok

A titkos szöveg:

hzfry mv hqythvhd wkn xswzlh, ok qxexgvycfs qpxuxic vejgdnc, pr yodptkn, ji esw.

2. ábra De Vignére többábécés módszerének működése

a lenyomat az esetek jelentős részében nagyobb, mint az eredeti adat, a leképezéssel – leegyszerűsítve – egy nagyobb elemszámú halmaz elemeit képezzük le egy kisebb elemszámú halmazba. Elkerülhetetlen tehát, hogy a kiinduló halmaz több eleméhez is ugyanaz az eredmény tartozzon. Ezt nevezik lenyomatütközésnek.

Egyszerű titkosító algoritmusok

A korszerűbb, összetettebb megoldások megértéséhez szükség van az egyszerűbb eljárások és hibáik ismeretére. A titkosítás használatának kezdeteinél a tudományok nem álltak még a mai szinten. Az akkoriban feltörhetetlennek tűnő algoritmusokat a tudomány folyamatos fejlődésének eredményeképpen ma könnyedén fel tudjuk törni. A titkosítás elmélete és gyakorlata a matematika, az információelmélet és az algoritmuselmélet fejlődésével folyamatosan változik, fejlődik. A fontosabb korai titkosítási módszereknek és hibáiknak alapos megismerésével lényegesen jobban átlátható, hogy milyen típusú támadásokkal kell számolnia a titkosítással foglalkozóknak.

1. lista Mini titkosítónk: megatitok_3000.pl

```
#!/usr/bin/perl -w

sub help();
use Getopt::Std;
my %changetable = ( "Æ" => "a", "Ø" => "e",
↳ " " => "i", " " => "o",
↳ "o" => "o", " " => "o", "æ" => "u", "u"
=> "u", " " => "u",
↳ "`" => "a", " " => "e", "¨" => "i", " "
=> "o", "O" => "o",
↳ " " => "o", " " => "u", "Û" => "u", " "
=> "u" );

my %opts;
getopts("edp:", \%opts);
if (! defined($opts{p}))
{
    print("A jelsz sajnos nincs megadva.\n");
    help();
}
$pwd = $opts{p};
$plen = length($pwd);

if (! (defined($opts{e}) or defined($opts{d})))
{
    $opts{e}=1;
}
elsif (defined($opts{e}) and defined($opts{d}))
{
    print("Egyszerre nem tudok titkos tani
↳s visszafejteni.\n");
    help();
}

my $letters = 0;
while(<>)
{
    chomp;
    for ($i=0; $i<length; $i++)
    {
        my $l = lc(substr($_, $i, 1));
        if (defined($changetable{$l}))
        {
            $l = $changetable{$l};
        }

        if($l =~ /\w/)
            {
                if(defined($opts{e}))
                {
                    # encrypt
                    print(chr(((ord($l)-97 +
↳ord(substr($pwd, $letters % $plen,
↳1))-97) % 26)+97));
                }
                else
                {
                    # decrypt
                    my $char = ord($l)-97 -
↳ord(substr($pwd, $letters %
↳$plen, 1))+97;
                    if ($char < 0)
                    {
                        $char += 26;
                    }
                    print(chr($char+97));
                }
                $letters++;
            }
            else
            {
                print($l)
            }
        }
        print("\n");
    }

sub help()
{
    print <<EOF;
    HasznÆlat: de_vigenere -p <jelsz > [-e|-d]
    A program Blaise De VigenÆre titkos t
    algoritmusÆval titkos t vagy fejt
    vissza, a megadott jelsz val. A jelsz t
    a -p paramÆterben kell megadni, a
    titkos tÆshoz a -e, a visszafejtÆshez
    pedig a -d paramÆtert kell megadni.
    Ha egyik sincs megadva, akkor
    a program titkos t.

    EOF
    exit(1);
}
}
```

Julius Ceasar eljárása

A következő eljárást a történelemlkönyvek szerint *Julius Caesar* alkalmazta először, ezért róla nevezték el. Meglepő módon a neve: Ceasar-féle titkosítás. Lényege a következő: a nyílt szöveg betűit úgy titkosítjuk, hogy azokat az ábécé szerint megadott betűvel eltoljuk. Ha az eltolás négy betű, akkor a nyílt szöveg „a” betűjéből a kriptoszövegben „e” betű lesz, a „b”-ből „f”, és így tovább. Ennél az algoritmusnál a kulcs az eltolás mértéke, jelen példánkban a négy. Az így kapott kriptoszöveg első ránézésre betűk összefüggéstelen halmaza. Ezt az eljárást szemlélteti az 1. ábra. Az ábrán is jól

látható azonban ennek a módszernek a hibája. Mivel a nyílt szöveg bizonyos szabályosságokat mutat, a módszer viszonylag könnyen felismerhető, megfejthető. Minden nyelvre jellemző például a betűk előfordulási valószínűsége, a betű-többszöröződések, a betűkettősök és -hármások átlagos száma. Mivel maga az alapmódszer csak a betűk számának megfelelő eltolási lehetőséget kínál, a titok próbálgatással hamar kideríthető. Ha nem egyszerűen eltoljuk a betűket, hanem a helyettesítésnél valamilyen véletlenszerű sorrendet használunk, az a próbálgatásos feltérést kissé nehezebbé teszi (a lehetőségek száma: hozzávetőleg a 25 faktoriális).

	Kis ábécé szóköz nélkül	Kis ábécé szóközzel	Ékezetes betűk szóköz nélkül
A	11,55	10,07	9,35
Á	-	-	3,72
B	2,38	2,12	1,72
C	0,63	0,54	0,60
D	1,79	1,42	1,71
E	14,26	11,86	9,71
É	-	-	3,87
F	0,94	0,83	0,88
G	3,22	2,87	3,55
H	1,68	1,37	1,23
I	5,48	4,84	4,39
J	1,05	0,90	1,21
K	5,84	5,26	5,35
L	6,23	5,44	6,30
M	3,65	3,29	3,92
N	5,47	4,69	5,47
O	6,87	6,04	4,47
Ö	-	-	2,14
P	1,09	0,88	1,04
Q	0,00	0,00	0,00
R	3,76	3,23	4,22
S	5,89	5,10	6,57
T	7,35	6,12	7,87
U	2,47	2,24	1,29
Ü	-	-	0,93
V	1,66	1,42	1,81
W	0,00	0,00	0,00
X	0,02	0,01	0,01
Y	1,92	1,64	2,21
Z	4,79	4,14	4,46
Szóköz	-	13,70	-

A magyar nyelv betűgyakoriságai 10 000 betűs újságszöveg alapján

Ezt a módszert nevezik egyábécés titkosítási módszernek. A módszer használata mellett a küldő és a fogadó oldaláról a hozzárendelési táblázatnak ismertnek kell lennie (ebben az esetben a táblázat a kulcs). A módszer hibája, hogy nem fedi el a nyílt szöveg szabályszerűségeit. Mivel minden betűírásos nyelvben meghatározhatók a betűk előfordulási valószínűségei, a titokfejtőnek nem kell próbálgatással piszmognia. Nem kell mást tennie, mint meghatározni az egyes betűk (jelek) számát a kriptoszövegben, és azok előfordulási arányai alapján a titkosítási táblázat könnyen létrehozható.

De Vignére többábécés módszere

Többek közt a fent ismertetett eljárás néhány továbbfejlesztett változatát ismertette *Blaise De Vignére* a középkorban. Az ő eljárásának az a lényege, hogy a nyílt szöveg betűinek helyettesítése a szövegben elfoglalt helyüktől is függ. Ehhez egy betű-táblázatot állít fel, amely egymás alatt az újra és újra eggyel eltolt ábécét tartalmazza. Az eljárás működési elvét a 2. ábra és az 1. listán látható egyszerű kis Perl-program szemlélteti. A nyílt szöveg minden betűjének egy jelszóbetűt feleltet meg. Amennyiben a jelszó rövidebb, mint a titkosítandó szöveg, a jelszó ismételtetésével nyújtja meg a megfelelő hosszúságúra. A kriptoszöveg egy betűjét a táblázatban a nyílt szöveg megfelelő betűje és a hozzá tartozó jelszóbetű határozza meg. Ez a gyakorlatban azt jelenti, hogy a jelszó betűi határozzák meg, hogy az adott betű melyik eltolási táblázat szerint lesz módosítva. Ez az eljárás kismértékben képes elfedni a nyílt szöveg szabályszerűségeit, de megadott felhasználása sem védett a statisztikai alapú támadások ellen. Egyértelmű, hogy ha a titkos kulcsszót ismételtetve használjuk a betűk eltolásának meghatározására, akkor a betűk minden ismétlésnél a korábbival megegyezően lesznek eltolva. Ha tehát meg tudnánk határozni a titkos jelszó hosszát, akkor minden periódusra meghatározhatnánk a korábbiakban említett betűvalószínűségeket, így a nyílt szöveg – és ezáltal a jelszó – minden egyes betűjét meg tudnánk határozni. Ennek kivitelezése a következő: mivel a módszer a betűk előfordulási valószínűségét a kriptoszövegben egyenletesebbé teszi, ha meg tudnánk határozni, hogy a szöveg milyen periódusai térnek el leginkább az egyenletestől – tehát mely szakaszok mutatják a leginkább a természetes nyelv jellegzetességeit –, akkor megtudnánk a jelszó hosszát. Rendre meghatározzuk minden második, harmadik, negyedik stb. betű csoportjának betűeloszlását, és amelyik a legjobban egyezik a természetes nyelv betűeloszlásával, annál ellenőrizzük, hogy a periódus minden betűcsoportja a magyar nyelv betűvalószínűségeit hozza-e. A fenti elemzés csak akkor tud a megfelelő hatékonysággal működni, ha elegendő kriptoszöveg áll a rendelkezésünkre. Nagy előnye, hogy nemcsak ezt a klasszikus eljárást lehet hatékonyan törni vele, hanem egy általánosabb módszert is. Ha ugyanis a titkosító táblázat soraiban nem eltolt, hanem kevert betűk vannak, az egyes sorok betűsorrendjét is meg kell találnunk. Ezt pedig a jelszó minden betűjére el kell végeznünk. Ennél a módszernél lényegesen jobb védeltséget ad, ha az első, a jelszóval titkosított blokkot használjuk fel mint a soron következő blokk jelszavát és így tovább. Nagyon hasonló alapelven működnek a modern szimmetrikus algoritmusok is.

Mini titkosítónk

Az 1. listán látható példaprogram nagyon egyszerű megvalósítása a fent tárgyalt De Vignére-féle algoritmus egy formájának. Az egyszerűség kedvéért csak a betűket titkosítja vagy fejt vissza, az egyéb karaktereket figyelmen kívül hagyja. További egyszerűsítés, hogy minden betűt kisbetűssé alakít és az ékezetes betűket ékezetmentesekre cseréli le. Használata nagyon egyszerű. A program a -p kapcsoló után várja a titkosító jelszót, ha mást nem adunk meg, akkor titkosít. Ha a -d kapcsolót is megadjuk, akkor a megadott jelszó felhasználásával visszafejti a kriptoszöveget. A nyílt vagy kriptoszöveget a szabványos bemenetén várja, és a szabvány kimenetén jelenik meg az eredmény. Ez az egyszerű program így elég karcsú, de kis módosítással tökéletes titkosító készíthető belőle. Elegendő úgy átírni, hogy a jelszót egy állományból vegye. Az állományban lévő jelszónak természetesen meg kell felelnie a korábban leírt feltételeknek. További érdekesség, hogy ha egyetlen betűs

© Kiskapu Kft. Minden jog fenntartva

titkosító jelszót választunk, a korábban leírt Caesar-féle eljárást kapjuk, mivel minden betűt ugyanennyival tolnuk el.

A program használatával lelkes olvasóink érdekes feladat elé állíthatják magukat: írjanak programot a korábban ismertetett fejtési eljárás önműködővé tételére. Ennek megkönnyítésére *táblázatunkban* közöljük a magyar nyelv betűgyakoriságait, a következő forrás alapján: [fulop]. Vájt fülűek természetesen maguk is elkészíthetik a betűgyakorisági táblázatot az Interneten található bőséges magyar nyelvű írott anyag és egy egyszerű kis program segítségével. Az *Igazi Perl-guru* így oldaná meg:

```
perl -ne '@l=split(//);
foreach(@l){if(/[:alpha:]|[\000-\007]|[\010-\017]|[\020-\027]|[\030-\037]|[\040-\047]|[\050-\057]|[\060-\067]|[\070-\077]|[\080-\087]|[\090-\097]|[\0A0-\0A7]|[\0B0-\0B7]|[\0C0-\0C7]|[\0D0-\0D7]|[\0E0-\0E7]|[\0F0-\0F7]|[\0100-\0107]|[\0110-\0117]|[\0120-\0127]|[\0130-\0137]|[\0140-\0147]|[\0150-\0157]|[\0160-\0167]|[\0170-\0177]|[\0180-\0187]|[\0190-\0197]|[\01A0-\01A7]|[\01B0-\01B7]|[\01C0-\01C7]|[\01D0-\01D7]|[\01E0-\01E7]|[\01F0-\01F7]|[\0200-\0207]|[\0210-\0217]|[\0220-\0227]|[\0230-\0237]|[\0240-\0247]|[\0250-\0257]|[\0260-\0267]|[\0270-\0277]|[\0280-\0287]|[\0290-\0297]|[\02A0-\02A7]|[\02B0-\02B7]|[\02C0-\02C7]|[\02D0-\02D7]|[\02E0-\02E7]|[\02F0-\02F7]|[\0300-\0307]|[\0310-\0317]|[\0320-\0327]|[\0330-\0337]|[\0340-\0347]|[\0350-\0357]|[\0360-\0367]|[\0370-\0377]|[\0380-\0387]|[\0390-\0397]|[\03A0-\03A7]|[\03B0-\03B7]|[\03C0-\03C7]|[\03D0-\03D7]|[\03E0-\03E7]|[\03F0-\03F7]|[\0400-\0407]|[\0410-\0417]|[\0420-\0427]|[\0430-\0437]|[\0440-\0447]|[\0450-\0457]|[\0460-\0467]|[\0470-\0477]|[\0480-\0487]|[\0490-\0497]|[\04A0-\04A7]|[\04B0-\04B7]|[\04C0-\04C7]|[\04D0-\04D7]|[\04E0-\04E7]|[\04F0-\04F7]|[\0500-\0507]|[\0510-\0517]|[\0520-\0527]|[\0530-\0537]|[\0540-\0547]|[\0550-\0557]|[\0560-\0567]|[\0570-\0577]|[\0580-\0587]|[\0590-\0597]|[\05A0-\05A7]|[\05B0-\05B7]|[\05C0-\05C7]|[\05D0-\05D7]|[\05E0-\05E7]|[\05F0-\05F7]|[\0600-\0607]|[\0610-\0617]|[\0620-\0627]|[\0630-\0637]|[\0640-\0647]|[\0650-\0657]|[\0660-\0667]|[\0670-\0677]|[\0680-\0687]|[\0690-\0697]|[\06A0-\06A7]|[\06B0-\06B7]|[\06C0-\06C7]|[\06D0-\06D7]|[\06E0-\06E7]|[\06F0-\06F7]|[\0700-\0707]|[\0710-\0717]|[\0720-\0727]|[\0730-\0737]|[\0740-\0747]|[\0750-\0757]|[\0760-\0767]|[\0770-\0777]|[\0780-\0787]|[\0790-\0797]|[\07A0-\07A7]|[\07B0-\07B7]|[\07C0-\07C7]|[\07D0-\07D7]|[\07E0-\07E7]|[\07F0-\07F7]|[\0800-\0807]|[\0810-\0817]|[\0820-\0827]|[\0830-\0837]|[\0840-\0847]|[\0850-\0857]|[\0860-\0867]|[\0870-\0877]|[\0880-\0887]|[\0890-\0897]|[\08A0-\08A7]|[\08B0-\08B7]|[\08C0-\08C7]|[\08D0-\08D7]|[\08E0-\08E7]|[\08F0-\08F7]|[\0900-\0907]|[\0910-\0917]|[\0920-\0927]|[\0930-\0937]|[\0940-\0947]|[\0950-\0957]|[\0960-\0967]|[\0970-\0977]|[\0980-\0987]|[\0990-\0997]|[\09A0-\09A7]|[\09B0-\09B7]|[\09C0-\09C7]|[\09D0-\09D7]|[\09E0-\09E7]|[\09F0-\09F7]|[\0A00-\0A07]|[\0A10-\0A17]|[\0A20-\0A27]|[\0A30-\0A37]|[\0A40-\0A47]|[\0A50-\0A57]|[\0A60-\0A67]|[\0A70-\0A77]|[\0A80-\0A87]|[\0A90-\0A97]|[\0AA0-\0AA7]|[\0AB0-\0AB7]|[\0AC0-\0AC7]|[\0AD0-\0AD7]|[\0AE0-\0AE7]|[\0AF0-\0AF7]|[\0B00-\0B07]|[\0B10-\0B17]|[\0B20-\0B27]|[\0B30-\0B37]|[\0B40-\0B47]|[\0B50-\0B57]|[\0B60-\0B67]|[\0B70-\0B77]|[\0B80-\0B87]|[\0B90-\0B97]|[\0BA0-\0BA7]|[\0BB0-\0BB7]|[\0BC0-\0BC7]|[\0BD0-\0BD7]|[\0BE0-\0BE7]|[\0BF0-\0BF7]|[\0C00-\0C07]|[\0C10-\0C17]|[\0C20-\0C27]|[\0C30-\0C37]|[\0C40-\0C47]|[\0C50-\0C57]|[\0C60-\0C67]|[\0C70-\0C77]|[\0C80-\0C87]|[\0C90-\0C97]|[\0CA0-\0CA7]|[\0CB0-\0CB7]|[\0CC0-\0CC7]|[\0CD0-\0CD7]|[\0CE0-\0CE7]|[\0CF0-\0CF7]|[\0D00-\0D07]|[\0D10-\0D17]|[\0D20-\0D27]|[\0D30-\0D37]|[\0D40-\0D47]|[\0D50-\0D57]|[\0D60-\0D67]|[\0D70-\0D77]|[\0D80-\0D87]|[\0D90-\0D97]|[\0DA0-\0DA7]|[\0DB0-\0DB7]|[\0DC0-\0DC7]|[\0DD0-\0DD7]|[\0DE0-\0DE7]|[\0DF0-\0DF7]|[\0E00-\0E07]|[\0E10-\0E17]|[\0E20-\0E27]|[\0E30-\0E37]|[\0E40-\0E47]|[\0E50-\0E57]|[\0E60-\0E67]|[\0E70-\0E77]|[\0E80-\0E87]|[\0E90-\0E97]|[\0EA0-\0EA7]|[\0EB0-\0EB7]|[\0EC0-\0EC7]|[\0ED0-\0ED7]|[\0EE0-\0EE7]|[\0EF0-\0EF7]|[\0F00-\0F07]|[\0F10-\0F17]|[\0F20-\0F27]|[\0F30-\0F37]|[\0F40-\0F47]|[\0F50-\0F57]|[\0F60-\0F67]|[\0F70-\0F77]|[\0F80-\0F87]|[\0F90-\0F97]|[\0FA0-\0FA7]|[\0FB0-\0FB7]|[\0FC0-\0FC7]|[\0FD0-\0FD7]|[\0FE0-\0FE7]|[\0FF0-\0FF7]|[\01000-\01007]|[\010010-\010017]|[\010020-\010027]|[\010030-\010037]|[\010040-\010047]|[\010050-\010057]|[\010060-\010067]|[\010070-\010077]|[\010080-\010087]|[\010090-\010097]|[\0100A0-\0100A7]|[\0100B0-\0100B7]|[\0100C0-\0100C7]|[\0100D0-\0100D7]|[\0100E0-\0100E7]|[\0100F0-\0100F7]|[\010100-\010107]|[\010110-\010117]|[\010120-\010127]|[\010130-\010137]|[\010140-\010147]|[\010150-\010157]|[\010160-\010167]|[\010170-\010177]|[\010180-\010187]|[\010190-\010197]|[\0101A0-\0101A7]|[\0101B0-\0101B7]|[\0101C0-\0101C7]|[\0101D0-\0101D7]|[\0101E0-\0101E7]|[\0101F0-\0101F7]|[\010200-\010207]|[\010210-\010217]|[\010220-\010227]|[\010230-\010237]|[\010240-\010247]|[\010250-\010257]|[\010260-\010267]|[\010270-\010277]|[\010280-\010287]|[\010290-\010297]|[\0102A0-\0102A7]|[\0102B0-\0102B7]|[\0102C0-\0102C7]|[\0102D0-\0102D7]|[\0102E0-\0102E7]|[\0102F0-\0102F7]|[\010300-\010307]|[\010310-\010317]|[\010320-\010327]|[\010330-\010337]|[\010340-\010347]|[\010350-\010357]|[\010360-\010367]|[\010370-\010377]|[\010380-\010387]|[\010390-\010397]|[\0103A0-\0103A7]|[\0103B0-\0103B7]|[\0103C0-\0103C7]|[\0103D0-\0103D7]|[\0103E0-\0103E7]|[\0103F0-\0103F7]|[\010400-\010407]|[\010410-\010417]|[\010420-\010427]|[\010430-\010437]|[\010440-\010447]|[\010450-\010457]|[\010460-\010467]|[\010470-\010477]|[\010480-\010487]|[\010490-\010497]|[\0104A0-\0104A7]|[\0104B0-\0104B7]|[\0104C0-\0104C7]|[\0104D0-\0104D7]|[\0104E0-\0104E7]|[\0104F0-\0104F7]|[\010500-\010507]|[\010510-\010517]|[\010520-\010527]|[\010530-\010537]|[\010540-\010547]|[\010550-\010557]|[\010560-\010567]|[\010570-\010577]|[\010580-\010587]|[\010590-\010597]|[\0105A0-\0105A7]|[\0105B0-\0105B7]|[\0105C0-\0105C7]|[\0105D0-\0105D7]|[\0105E0-\0105E7]|[\0105F0-\0105F7]|[\010600-\010607]|[\010610-\010617]|[\010620-\010627]|[\010630-\010637]|[\010640-\010647]|[\010650-\010657]|[\010660-\010667]|[\010670-\010677]|[\010680-\010687]|[\010690-\010697]|[\0106A0-\0106A7]|[\0106B0-\0106B7]|[\0106C0-\0106C7]|[\0106D0-\0106D7]|[\0106E0-\0106E7]|[\0106F0-\0106F7]|[\010700-\010707]|[\010710-\010717]|[\010720-\010727]|[\010730-\010737]|[\010740-\010747]|[\010750-\010757]|[\010760-\010767]|[\010770-\010777]|[\010780-\010787]|[\010790-\010797]|[\0107A0-\0107A7]|[\0107B0-\0107B7]|[\0107C0-\0107C7]|[\0107D0-\0107D7]|[\0107E0-\0107E7]|[\0107F0-\0107F7]|[\010800-\010807]|[\010810-\010817]|[\010820-\010827]|[\010830-\010837]|[\010840-\010847]|[\010850-\010857]|[\010860-\010867]|[\010870-\010877]|[\010880-\010887]|[\010890-\010897]|[\0108A0-\0108A7]|[\0108B0-\0108B7]|[\0108C0-\0108C7]|[\0108D0-\0108D7]|[\0108E0-\0108E7]|[\0108F0-\0108F7]|[\010900-\010907]|[\010910-\010917]|[\010920-\010927]|[\010930-\010937]|[\010940-\010947]|[\010950-\010957]|[\010960-\010967]|[\010970-\010977]|[\010980-\010987]|[\010990-\010997]|[\0109A0-\0109A7]|[\0109B0-\0109B7]|[\0109C0-\0109C7]|[\0109D0-\0109D7]|[\0109E0-\0109E7]|[\0109F0-\0109F7]|[\010A00-\010A07]|[\010A10-\010A17]|[\010A20-\010A27]|[\010A30-\010A37]|[\010A40-\010A47]|[\010A50-\010A57]|[\010A60-\010A67]|[\010A70-\010A77]|[\010A80-\010A87]|[\010A90-\010A97]|[\010AA0-\010AA7]|[\010AB0-\010AB7]|[\010AC0-\010AC7]|[\010AD0-\010AD7]|[\010AE0-\010AE7]|[\010AF0-\010AF7]|[\010B00-\010B07]|[\010B10-\010B17]|[\010B20-\010B27]|[\010B30-\010B37]|[\010B40-\010B47]|[\010B50-\010B57]|[\010B60-\010B67]|[\010B70-\010B77]|[\010B80-\010B87]|[\010B90-\010B97]|[\010BA0-\010BA7]|[\010BB0-\010BB7]|[\010BC0-\010BC7]|[\010BD0-\010BD7]|[\010BE0-\010BE7]|[\010BF0-\010BF7]|[\010C00-\010C07]|[\010C10-\010C17]|[\010C20-\010C27]|[\010C30-\010C37]|[\010C40-\010C47]|[\010C50-\010C57]|[\010C60-\010C67]|[\010C70-\010C77]|[\010C80-\010C87]|[\010C90-\010C97]|[\010CA0-\010CA7]|[\010CB0-\010CB7]|[\010CC0-\010CC7]|[\010CD0-\010CD7]|[\010CE0-\010CE7]|[\010CF0-\010CF7]|[\010D00-\010D07]|[\010D10-\010D17]|[\010D20-\010D27]|[\010D30-\010D37]|[\010D40-\010D47]|[\010D50-\010D57]|[\010D60-\010D67]|[\010D70-\010D77]|[\010D80-\010D87]|[\010D90-\010D97]|[\010DA0-\010DA7]|[\010DB0-\010DB7]|[\010DC0-\010DC7]|[\010DD0-\010DD7]|[\010DE0-\010DE7]|[\010DF0-\010DF7]|[\010E00-\010E07]|[\010E10-\010E17]|[\010E20-\010E27]|[\010E30-\010E37]|[\010E40-\010E47]|[\010E50-\010E57]|[\010E60-\010E67]|[\010E70-\010E77]|[\010E80-\010E87]|[\010E90-\010E97]|[\010EA0-\010EA7]|[\010EB0-\010EB7]|[\010EC0-\010EC7]|[\010ED0-\010ED7]|[\010EE0-\010EE7]|[\010EF0-\010EF7]|[\010F00-\010F07]|[\010F10-\010F17]|[\010F20-\010F27]|[\010F30-\010F37]|[\010F40-\010F47]|[\010F50-\010F57]|[\010F60-\010F67]|[\010F70-\010F77]|[\010F80-\010F87]|[\010F90-\010F97]|[\010FA0-\010FA7]|[\010FB0-\010FB7]|[\010FC0-\010FC7]|[\010FD0-\010FD7]|[\010FE0-\010FE7]|[\010FF0-\010FF7]|[\011000-\011007]|[\011010-\011017]|[\011020-\011027]|[\011030-\011037]|[\011040-\011047]|[\011050-\011057]|[\011060-\011067]|[\011070-\011077]|[\011080-\011087]|[\011090-\011097]|[\0110A0-\0110A7]|[\0110B0-\0110B7]|[\0110C0-\0110C7]|[\0110D0-\0110D7]|[\0110E0-\0110E7]|[\0110F0-\0110F7]|[\011100-\011107]|[\011110-\011117]|[\011120-\011127]|[\011130-\011137]|[\011140-\011147]|[\011150-\011157]|[\011160-\011167]|[\011170-\011177]|[\011180-\011187]|[\011190-\011197]|[\0111A0-\0111A7]|[\0111B0-\0111B7]|[\0111C0-\0111C7]|[\0111D0-\0111D7]|[\0111E0-\0111E7]|[\0111F0-\0111F7]|[\011200-\011207]|[\011210-\011217]|[\011220-\011227]|[\011230-\011237]|[\011240-\011247]|[\011250-\011257]|[\011260-\011267]|[\011270-\011277]|[\011280-\011287]|[\011290-\011297]|[\0112A0-\0112A7]|[\0112B0-\0112B7]|[\0112C0-\0112C7]|[\0112D0-\0112D7]|[\0112E0-\0112E7]|[\0112F0-\0112F7]|[\011300-\011307]|[\011310-\011317]|[\011320-\011327]|[\011330-\011337]|[\011340-\011347]|[\011350-\011357]|[\011360-\011367]|[\011370-\011377]|[\011380-\011387]|[\011390-\011397]|[\0113A0-\0113A7]|[\0113B0-\0113B7]|[\0113C0-\0113C7]|[\0113D0-\0113D7]|[\0113E0-\0113E7]|[\0113F0-\0113F7]|[\011400-\011407]|[\011410-\011417]|[\011420-\011427]|[\011430-\011437]|[\011440-\011447]|[\011450-\011457]|[\011460-\011467]|[\011470-\011477]|[\011480-\011487]|[\011490-\011497]|[\0114A0-\0114A7]|[\0114B0-\0114B7]|[\0114C0-\0114C7]|[\0114D0-\0114D7]|[\0114E0-\0114E7]|[\0114F0-\0114F7]|[\011500-\011507]|[\011510-\011517]|[\011520-\011527]|[\011530-\011537]|[\011540-\011547]|[\011550-\011557]|[\011560-\011567]|[\011570-\011577]|[\011580-\011587]|[\011590-\011597]|[\0115A0-\0115A7]|[\0115B0-\0115B7]|[\0115C0-\0115C7]|[\0115D0-\0115D7]|[\0115E0-\0115E7]|[\0115F0-\0115F7]|[\011600-\011607]|[\011610-\011617]|[\011620-\011627]|[\011630-\011637]|[\011640-\011647]|[\011650-\011657]|[\011660-\011667]|[\011670-\011677]|[\011680-\011687]|[\011690-\011697]|[\0116A0-\0116A7]|[\0116B0-\0116B7]|[\0116C0-\0116C7]|[\0116D0-\0116D7]|[\0116E0-\0116E7]|[\0116F0-\0116F7]|[\011700-\011707]|[\011710-\011717]|[\011720-\011727]|[\011730-\011737]|[\011740-\011747]|[\011750-\011757]|[\011760-\011767]|[\011770-\011777]|[\011780-\011787]|[\011790-\011797]|[\0117A0-\0117A7]|[\0117B0-\0117B7]|[\0117C0-\0117C7]|[\0117D0-\0117D7]|[\0117E0-\0117E7]|[\0117F0-\0117F7]|[\011800-\011807]|[\011810-\011817]|[\011820-\011827]|[\011830-\011837]|[\011840-\011847]|[\011850-\011857]|[\011860-\011867]|[\011870-\011877]|[\011880-\011887]|[\011890-\011897]|[\0118A0-\0118A7]|[\0118B0-\0118B7]|[\0118C0-\0118C7]|[\0118D0-\0118D7]|[\0118E0-\0118E7]|[\0118F0-\0118F7]|[\011900-\011907]|[\011910-\011917]|[\011920-\011927]|[\011930-\011937]|[\011940-\011947]|[\011950-\011957]|[\011960-\011967]|[\011970-\011977]|[\011980-\011987]|[\011990-\011997]|[\0119A0-\0119A7]|[\0119B0-\0119B7]|[\0119C0-\0119C7]|[\0119D0-\0119D7]|[\0119E0-\0119E7]|[\0119F0-\0119F7]|[\011A00-\011A07]|[\011A10-\011A17]|[\011A20-\011A27]|[\011A30-\011A37]|[\011A40-\011A47]|[\011A50-\011A57]|[\011A60-\011A67]|[\011A70-\011A77]|[\011A80-\011A87]|[\011A90-\011A97]|[\011AA0-\011AA7]|[\011AB0-\011AB7]|[\011AC0-\011AC7]|[\011AD0-\011AD7]|[\011AE0-\011AE7]|[\011AF0-\011AF7]|[\011B00-\011B07]|[\011B10-\011B17]|[\011B20-\011B27]|[\011B30-\011B37]|[\011B40-\011B47]|[\011B50-\011B57]|[\011B60-\011B67]|[\011B70-\011B77]|[\011B80-\011B87]|[\011B90-\011B97]|[\011BA0-\011BA7]|[\011BB0-\011BB7]|[\011BC0-\011BC7]|[\011BD0-\011BD7]|[\011BE0-\011BE7]|[\011BF0-\011BF7]|[\011C00-\011C07]|[\011C10-\011C17]|[\011C20-\011C27]|[\011C30-\011C37]|[\011C40-\011C47]|[\011C50-\011C57]|[\011C60-\011C67]|[\011C70-\011C77]|[\011C80-\011C87]|[\011C90-\011C97]|[\011CA0-\011CA7]|[\011CB0-\011CB7]|[\011CC0-\011CC7]|[\011CD0-\011CD7]|[\011CE0-\011CE7]|[\011CF0-\011CF7]|[\011D00-\011D07]|[\011D10-\011D17]|[\011D20-\011D27]|[\011D30-\011D37]|[\011D40-\011D47]|[\011D50-\011D57]|[\011D60-\011D67]|[\011D70-\011D77]|[\011D80-\011D87]|[\011D90-\011D97]|[\011DA0-\011DA7]|[\011DB0-\011DB7]|[\011DC0-\011DC7]|[\011DD0-\011DD7]|[\011DE0-\011DE7]|[\011DF0-\011DF7]|[\011E00-\011E07]|[\011E10-\011E17]|[\011E20-\011E27]|[\011E30-\011E37]|[\011E40-\011E47]|[\011E50-\011E57]|[\011E60-\011E67]|[\011E70-\011E77]|[\011E80-\011E87]|[\011E90-\011E97]|[\011EA0-\011EA7]|[\011EB0-\011EB7]|[\011EC0-\011EC7]|[\011ED0-\011ED7]|[\011EE0-\011EE7]|[\011EF0-\011EF7]|[\011F00-\011F07]|[\011F10-\011F17]|[\011F20-\011F27]|[\011F30-\011F37]|[\011F40-\011F47]|[\011F50-\011F57]|[\011F60-\011F67]|[\011F70-\011F77]|[\011F80-\011F87]|[\011F90-\011F97]|[\011FA0-\011FA7]|[\011FB0-\011FB7]|[\011FC0-\011FC7]|[\011FD0-\011FD7]|[\011FE0-\011FE7]|[\011FF0-\011FF7]|[\012000-\012007]|[\012010-\012017]|[\012020-\012027]|[\012030-\012037]|[\012040-\012047]|[\012050-\012057]|[\012060-\012067]|[\012070-\012077]|[\012080-\012087]|[\012090-\012097]|[\0120A0-\0120A7]|[\0120B0-\0120B7]|[\0120C0-\0120C7]|[\0120D0-\0120D7]|[\0120E0-\0120E7]|[\0120F0-\0120F7]|[\012100-\012107]|[\012110-\012117]|[\012120-\012127]|[\012130-\012137]|[\012140-\012147]|[\012150-\012157]|[\012160-\012167]|[\012170-\012177]|[\012180-\012187]|[\012190-\012197]|[\0121A0-\0121A7]|[\0121B0-\0121B7]|[\0121C0-\0121C7]|[\0121D0-\0121D7]|[\0121E0-\0121E7]|[\0121F0-\0121F7]|[\012200-\012207]|[\012210-\012217]|[\012220-\012227]|[\012230-\012237]|[\012240-\012247]|[\012250-\012257]|[\012260-\012267]|[\012270-\012277]|[\012280-\012287]|[\012290-\012297]|[\0122A0-\0122A7]|[\0122B0-\0122B7]|[\0122C0-\0122C7]|[\0122D0-\0122D7]|[\0122E0-\0122E7]|[\0122F0-\0122F7]|[\012300-\012307]|[\012310-\012317]|[\012320-\012327]|[\012330-\012337]|[\012340-\012347]|[\012350-\012357]|[\012360-\012367]|[\012370-\012377]|[\012380-\012387]|[\012390-\012397]|[\0123A0-\0123A7]|[\0123B0-\0123B7]|[\0123C0-\0123C7]|[\0123D0-\0123D7]|[\0123E0-\0123E7]|[\0123F0-\0123F7]|[\012400-\012407]|[\012410-\012417]|[\012420-\012427]|[\012430-\012437]|[\012440-\012447]|[\012450-\012457]|[\012460-\012467]|[\012470-\012477]|[\012480-\012487]|[\012490-\012497]|[\0124A0-\0124A7]|[\0124B0-\0124B7]|[\0124C0-\0124C7]|[\0124D0-\0124D7]|[\0124E0-\0124E7]|[\0124F0-\0124F7]|[\012500-\012507]|[\012510-\012517]|[\012520-\012527]|[\012530-\012537]|[\012540-\012547]|[\012550-\012557]|[\012560-\012567]|[\012570-\012577]|[\012580-\012587]|[\012590-\012597]|[\0125A0-\0125A7]|[\0125B0-\0125B7]|[\0125C0-\0125C7]|[\0125D0-\0125D7]|[\0125E0-\0125E7]|[\0125F0-\0125F7]|[\012600-\012607]|[\012610-\012617]|[\012620-\012627]|[\012630-\012637]|[\012640-\012647]|[\012650-\012657]|[\012660-\012667]|[\012670-\012677]|[\012680-\012687]|[\012690-\012697]|[\0126A0-\0126A7]|[\0126B0-\0126B7]|[\0126C0-\0126C7]|[\0126D0-\0126D7]|[\0126E0-\0126E7]|[\0126F0-\0126F7]|[\012700-\012707]|[\012710-\012717]|[\012720-\012727]|[\012730-\012737]|[\0
```