Funkcja stworzona na bazie Blake3.

Argumenty funkcji bazowej to stan (16 bajtów) i blok (32 bajtów). Wartością funkcji jest stan (16 bajtów).

Stan dzielimy na 8 liczb 16-bitowych (kolejność big endian) $\mathtt{w}[0], \ldots, \mathtt{w}[7]$.

Blok dzielimy na ciąg 16 liczb 16-bitowych (kolejność big endian) $\mathtt{m}[0], \ldots, \mathtt{m}[15]$.

W opisie algorytmu wszystkie liczby stanu i bloku zapisane są w systemie szesnastkowym. Funkcja będzie używała operacji na liczbach 16-bitowych:

- $\oplus$ - bitowy xor,
- $+$ dodawanie $\pmod{2^{16}}$,
- **rol** przesunięcie cykliczne w lewo.

Najpierw stan wejściowy $\mathtt{w}$ zamieniany jest na macierz $\mathtt{v}$ 4×4 liczb 16-bitowych tak, że

- $\mathtt{v}[0][\mathtt{i}] = \mathtt{w}[\mathtt{i}]$ dla $\mathtt{i} = 0 \ldots 4$,
- $\mathtt{v}[1][\mathtt{i}] = \mathtt{w}[\mathtt{i}+4]$ dla $\mathtt{i} = 0 \ldots 4$,
- $\mathtt{v}[2][0] = \mathtt{03F4}$, $\mathtt{v}[2][1] = \mathtt{774C}$, $\mathtt{v}[2][2] = \mathtt{5690}$, $\mathtt{v}[2][3] = \mathtt{C878}$
- $\mathtt{v}[3][0] = 0$, $\mathtt{v}[3][1]$ to numer bloku (zaczynając od 0), $\mathtt{v}[3][2] = \mathtt{v}[3][3] = 0$.

Algorytm składa się z 6 rund. Runda składa się z przekształceń pionowych (4 pierwsze przekształcenia $\mathtt{G}$) oraz przekształceń ukośnych (4 kolejne przekształcenia). Każda runda ma postać:

$\mathtt{G}(\mathtt{v}[0][0], \mathtt{v}[1][0], \mathtt{v}[2][0], \mathtt{v}[3][0], \mathtt{m}[\ 0], \mathtt{m}[\ 1])$
$\mathtt{G}(\mathtt{v}[0][1], \mathtt{v}[1][1], \mathtt{v}[2][1], \mathtt{v}[3][1], \mathtt{m}[\ 2], \mathtt{m}[\ 3])$
$\mathtt{G}(\mathtt{v}[0][2], \mathtt{v}[1][2], \mathtt{v}[2][2], \mathtt{v}[3][2], \mathtt{m}[\ 4], \mathtt{m}[\ 5])$
$\mathtt{G}(\mathtt{v}[0][3], \mathtt{v}[1][3], \mathtt{v}[2][3], \mathtt{v}[3][3], \mathtt{m}[\ 6], \mathtt{m}[\ 7])$
$\mathtt{G}(\mathtt{v}[0][0], \mathtt{v}[1][1], \mathtt{v}[2][2], \mathtt{v}[3][3], \mathtt{m}[\ 8], \mathtt{m}[\ 9])$
$\mathtt{G}(\mathtt{v}[0][1], \mathtt{v}[1][2], \mathtt{v}[2][3], \mathtt{v}[3][0], \mathtt{m}[10], \mathtt{m}[12])$
$\mathtt{G}(\mathtt{v}[0][2], \mathtt{v}[1][3], \mathtt{v}[2][0], \mathtt{v}[3][1], \mathtt{m}[12], \mathtt{m}[13])$
$\mathtt{G}(\mathtt{v}[0][3], \mathtt{v}[1][0], \mathtt{v}[2][1], \mathtt{v}[3][2], \mathtt{m}[14], \mathtt{m}[15])$
zamień wartości tablicy $\mathtt{m}$ według permutacji $\mathtt{s}$

gdzie permutacja $\mathtt{s}$ to

$$\mathtt{s} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 2 & 6 & 3 & 10 & 7 & 0 & 4 & 13 & 1 & 11 & 12 & 5 & 9 & 14 & 15 & 8 \end{pmatrix}$$

natomiast funkcja pomocnicza zmieniająca wartości $\mathtt{a}\,\mathtt{b}\,\mathtt{c}\,\mathtt{d}$, to

$\mathtt{G}(\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{x}, \mathtt{y})$
$\quad \mathtt{a} = \mathtt{a} + \mathtt{b} + \mathtt{x}$
$\quad \mathtt{d} = (\mathtt{d} \oplus \mathtt{a})\ \mathbf{rol}\ 3$
$\quad \mathtt{c} = \mathtt{c} + \mathtt{d}$
$\quad \mathtt{b} = (\mathtt{b} \oplus \mathtt{c})\ \mathbf{rol}\ 11$
$\quad \mathtt{a} = \mathtt{a} + \mathtt{b} + \mathtt{y}$
$\quad \mathtt{d} = (\mathtt{d} \oplus \mathtt{a})\ \mathbf{rol}\ 2$
$\quad \mathtt{c} = \mathtt{c} + \mathtt{d}$
$\quad \mathtt{b} = (\mathtt{b} \oplus \mathtt{c})\ \mathbf{rol}\ 5$

Wartością zwracaną jest $\mathtt{w}[\mathtt{i}] = \mathtt{w}[\mathtt{i}] \oplus \mathtt{v}[0][\mathtt{i}] \oplus \mathtt{v}[2][\mathtt{i}]$ i $\mathtt{w}[\mathtt{i}+4] = \mathtt{w}[\mathtt{i}+4] \oplus \mathtt{v}[1][\mathtt{i}] \oplus \mathtt{v}[3][\mathtt{i}]$ dla $\mathtt{i} = 0 \ldots 3$

Inicjalna wartość stanu to $\mathtt{w}[\mathtt{i}] = 0$ dla $\mathtt{i} = 0 \ldots 7$. Jeśli tekst jawny składa się z kilku bloków, to stan poprzedniego bloku jest stanem wejściowym dla kolejnego bloku.

Funkcją skrótu jest stan otrzymany z ostatniego bloku (big endian).

Przykład
Pierwszy blok wejściowy:

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F

Dane początkowe m

i	0	1	2	3	4	5	6	7
m	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F
i	8	9	10	11	12	13	14	15
m	1011	1213	1415	1617	1819	1A1B	1C1D	1E1F

Po pierwszej rundzie

Pionowe przekształcenia				
$i\backslash j$	0	1	2	3
0	E223	AEC7	D6CA	1B63
1	968D	F12E	8A86	8942
2	8CAB	D332	F0E2	150D
3	88AF	3BBE	5A0A	EC2D

Ukośne przekształcenia				
$i\backslash j$	0	1	2	3
0	9A48	51C8	CB46	0B5B
1	C467	19C9	8B75	DFC7
2	07F8	210C	EC1D	4214
3	FE99	5E73	AD9E	80C3

Dane m po pierwszej permutacji

i	0	1	2	3	4	5	6	7
m	0A0B	1011	0001	0405	0C0D	1617	0203	0809
i	8	9	10	11	12	13	14	15
m	1E1F	1819	0607	1213	1415	0E0F	1A1B	1C1D

Po drugiej rundzie

Pionowe przekształcenia				
$i\backslash j$	0	1	2	3
0	14B6	55E5	B04C	F8A2
1	8ABA	B0CB	BA74	9F21
2	4FBE	BBC8	92BE	0F8D
3	96AA	EBB3	2BEB	5E46

Ukośne przekształcenia				
$i\backslash j$	0	1	2	3
0	BD42	D5EF	CA04	7964
1	2C83	4B21	3599	E3FC
2	BF6C	A6B4	CBD0	64D0
3	4EEF	2E22	3607	49DD

Po trzeciej rundzie

Pionowe przekształcenia				
$i\backslash j$	0	1	2	3
0	7127	17F6	9997	E4CC
1	1B0F	B5C4	C4B3	74DF
2	2BF4	C974	1BC1	90E8
3	E2EB	F8F8	92B8	06B7

Ukośne przekształcenia				
$i\backslash j$	0	1	2	3
0	5700	1404	03AD	66FD
1	6AC1	48F9	1FB4	2179
2	2E2C	35AE	35A4	BBC0
3	18B0	9E09	71A6	EBE9

Po czwartej rundzie

Pionowe przekształcenia				
$i\backslash j$	0	1	2	3
0	8F0D	A3D2	DE2A	80CE
1	A3A8	A4FD	9EDF	8C61
2	DA01	CB94	6E69	D455
3	50AF	2CF7	2223	ACB3

Ukośne przekształcenia				
$i\backslash j$	0	1	2	3
0	3D6A	6EA6	4DA7	1F7C
1	0162	7558	34E9	9756
2	67B0	F4E5	2A94	5E78
3	F93B	7A75	B671	A774

Po piątej rundzie

Pionowe przekształcenia				
$i\backslash j$	0	1	2	3
0	B03E	9592	236F	0FE0
1	DD43	71D7	D52A	9436
2	21D4	32E4	3802	9841
3	3B2F	DADB	567E	8CEE

Ukośne przekształcenia				
$i\backslash j$	0	1	2	3
0	37A5	C8E3	0275	D4B9
1	6177	ACCC	0397	9E8F
2	3438	688B	661A	F609
3	DE86	A679	BB32	47C3

Po szóstej rundzie

Pionowe przekształcenia				
$i\backslash j$	0	1	2	3
0	E0C4	6331	319B	B22E
1	4E9C	7440	AFC3	23DC
2	C7F0	C0F4	1415	E7B2
3	F5FD	FAE0	E51F	9CA3

Ukośne przekształcenia				
$i\backslash j$	0	1	2	3
0	B7BC	8815	3E19	CC62
1	DD4E	C36A	DE1A	3671
2	4735	CB62	0CB5	8DF5
3	BE8D	5430	CBD7	EB2A

Wynikiem funkcji skrótu jest ciąg F0 89 43 77 32 AC 41 97 63 C3 97 5A 15 CD DD 5B.

Dopełnienie do pełnego bloku.

Do ciągu bajtów wiadomości zawsze należy dodać bajty dopełnienia, żeby minimalnie dopełnić wiadomość do wielokrotności długości bloku. Jeśli długość wiadomości jest wielokrotnością długości bloku, to należy dodać dodatkowy blok zawierający wyłącznie dopełnienie.

Dopełnienie, które należy wykorzystać to dopisanie na końcu wiadomości pojedynczego bajtu `7F` po czym dopisanie minimalnej liczby bajtów `FF` do dopełnienia bloku. Bajty `FF` mogą nie występować.

Przykłady dopełnienia dla bloku o długości 8 bajtów

wiadomość	wiadomość po dopełnieniu
pusta wiadomość	7F FF FF FF FF FF FF FF
7F	7F 7F FF FF FF FF FF FF
00 01 02 03	00 01 02 03 7F FF FF FF
00 01 02 03 04 05 06	00 01 02 03 04 05 06 7F
00 01 02 03 04 05 06 07	00 01 02 03 04 05 06 07 7F FF FF FF FF FF FF FF
00 01 02 03 04 05 06 07 08	00 01 02 03 04 05 06 07 08 7F FF FF FF FF FF FF

Zadanie

Dla podanej funkcji skrótu oraz podanego dopełnienia do pełnego bloku należy znaleźć dane wejściowe.

Wszystkie podane dane wejściowe są zapisane jako ciągi ASCII. Napis `a`*77 oznacza ciąg 77 znaków `a`. Poniżej przedstawione są przykładowe dane wejściowe i ich funkcje skrótu.

Dane wejściowe	funkcja skrótu
pusta wiadomość	89 8F E0 38 CC 44 AC 95 0F 78 F8 4D 87 96 98 C9
`AbCxYz`	E1 C1 3F 52 3C 78 75 89 22 FD 11 AA 31 32 D0 1C
`1234567890`	86 91 1F 68 BF 45 A5 D6 C2 95 B6 F7 95 D9 B9 BE
`Alamakota,kotmaale.`	B0 E3 5A D8 BC C3 0D 12 2F ED A6 09 DE 3C 99 1C
`Ty,ktorywchodzisz,zegnajsieznadzieja.`	86 2B EA 4A 83 77 CB 1C 7C F2 18 51 F7 29 D5 93
`Litwo,Ojczyznomoja!tyjestesjakzdrowie;`	94 FE 53 59 63 CD 40 55 AA 16 22 06 5A 34 55 A5
`a`*48000	73 8C 65 2D 72 74 EF C3 B8 F4 80 4C DC 2D 28 73
`a`*48479	37 05 B3 83 C5 F6 19 9B 87 4D D6 6A 8B B0 E7 49
`a`*48958	DB 87 B2 C0 C1 69 A7 85 96 E3 28 14 5B 46 BF AC

Należy znaleźć dane wejściowe dla podanych wyników funkcji skrótu:

Liczba znaków w danych wejściowych	funkcja skrótu
1	29 0D 8E 30 A7 F7 58 DE 02 3C 9C 74 62 33 63 1D
2	6C 34 6E 8D 30 67 EF 3B 7B C3 E5 C2 99 CC 75 35
3	E1 4D A6 D5 EB 17 15 BE CD 5D 46 80 D9 9D 6E DC
4	26 8D DE E3 CD 85 4D 73 80 E5 4F 61 57 12 86 CD
5	CF AC 55 48 46 A0 7C F5 54 34 4C 38 7B 8E 48 DC
6	22 AA 75 76 75 8A 39 78 77 BC 3A A0 40 F5 BD 12
7	62 07 25 80 37 4C 71 E6 0D 2C 83 5E 33 98 5B E5

dane zostały wygenerowane z poniższego zbioru 92 znaków
`qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890!@#$%^&*-_=+([{<)]}>'";:?,.\/|`