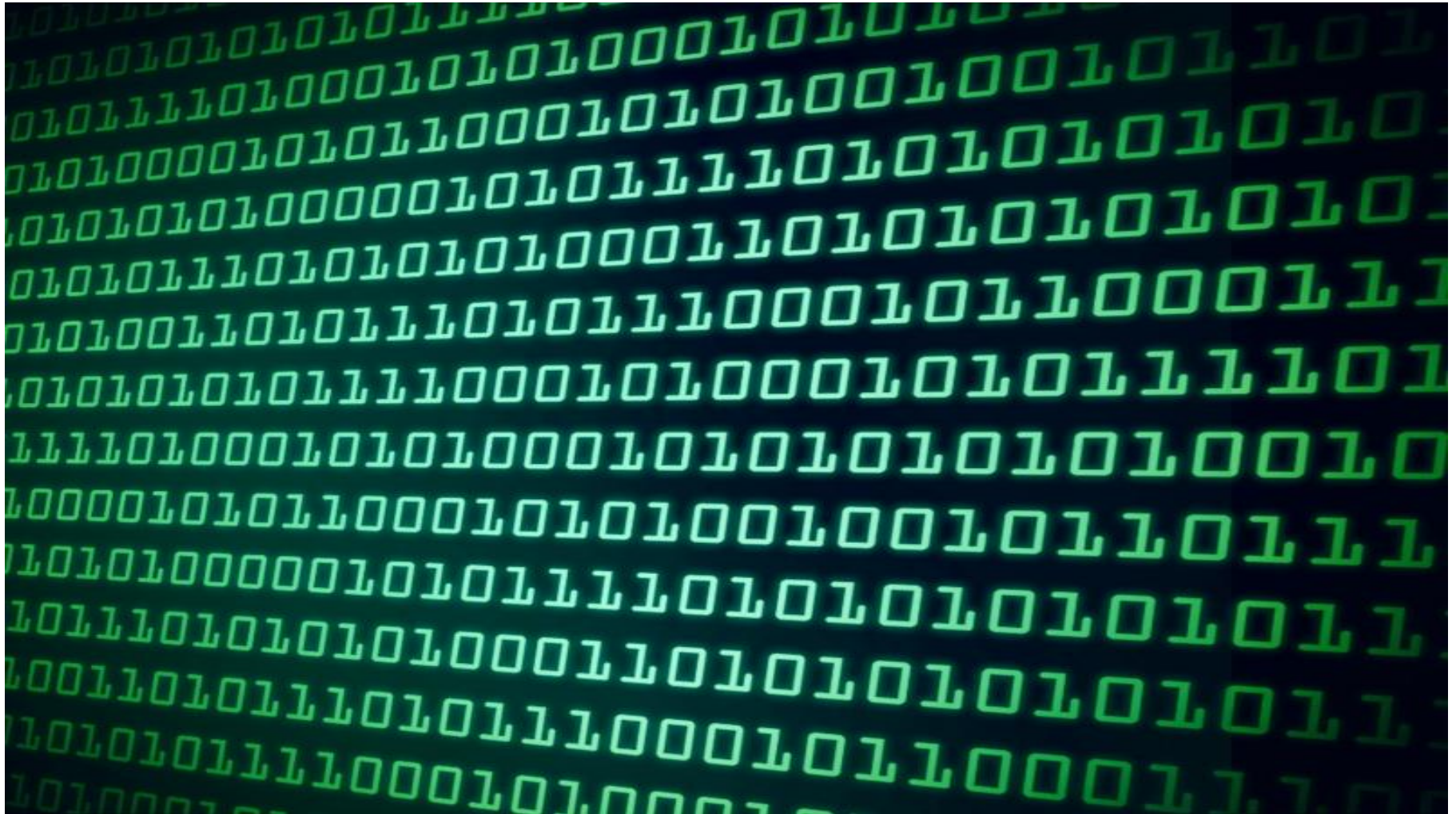


# Bitmasks



# Περιεχόμενα

.Εισαγωγή

.Bitwise Operators

.NOT

.AND

.OR

.XOR

.Αναπαράσταση

.Shift bits

.Turn on a bit

.Check a bit

.Turn off a bit

.Switch a bit

.Εύρεση τελευταίου bit

.Turn on n bits

# Εισαγωγή

Όταν ενώνουμε bitmasks μιλούμε για integers που χειριζόμαστε την δυαδική τους μορφή.

Με απλά λογία μιλούμε για πινάκες τύπου bool μεγέθους 64.

Γιατί bitmasks;

- Ελαφριά στην μνήμη
- Γρηγορότερα
- Λίγος κώδικας

# Bitwise Operators

Όπως υπάρχουν οι τελεστές (+, -, \*, \) που μας επιτρέπουν να ελέγχουμε αριθμούς στο Δεκαδικό σύστημα

Έτσι έχουμε και τελεστές που μας επιτρέπουν να ελέγχουμε αριθμούς στο δυαδικό σύστημα.

# NOT(~)

Όταν θέλουμε να πιάσουμε το αντίθετο ενός bit.

P	$\sim P$
0	1
1	0

# AND(&)

Όταν θέλουμε να δούμε αν και τα δυο bit είναι 1.

P	Q	P&Q
0	0	0
0	1	0
1	0	0
1	1	1

# OR(|)

Όταν θέλουμε να ξέρουμε αν τουλάχιστον ένα bit είναι 1.

P	Q	P Q
0	0	0
0	1	1
1	0	1
1	1	1

# XOR(^)

Όταν θέλουμε να δούμε αν μόνο ένα bit είναι 1.

P	Q	$P \wedge Q$
0	0	0
0	1	1
1	0	1
1	1	0



# Αναπαράσταση

Η αναπαράσταση των bitmasks γίνεται με την βοήθεια του δυαδικού συστήματος

Index from right to left	5	4	3	2	1	0
Power of 2	32	16	8	4	2	1
34 στο δυαδικό	1	0	0	0	1	0

# Shift bits

Όταν θέλουμε να μετακινήσουμε όλα τα bits  $N$  θέσεις αριστερά η δεξιά πολλαπλασιάζουμε η διαιρούμε με  $2^N$  ή απλώς χρησιμοποιούμε το  $\ll N$  ή  $\gg N$ .

$x = x \ll N;$	
10 σε δυαδικό	00001010
$10 \ll 2$	00101000

$x = x \gg N;$	
10 σε δυαδικό	00001010
$10 \gg 2$	00000010

!!Σημείωση: Όταν χρησιμοποιούμε bitmasks χρησιμοποιούμε τον τύπο unsigned long long

# Turn on a Bit

Όταν θέλουμε να μετατρέψουμε κάποιο συγκεκριμένο bit σε 1 το πετυχαίνουμε με την βοήθεια του OR.

x	34	00100010
y	$1 \ll 3$	00001000
$x = x   y$	42	00101010

# Check a Bit

Όταν θέλουμε να ελέγξουμε την τιμή ενός bit το κάνουμε με την βοήθεια του AND.

x	34	00100010
y	$1 \ll 3$	00001000
$x = x \& y$	0	00000000

x	34	00100010
y	$1 \ll 5$	00100000
$x = x \& y$	32	00100000

!!Σημείωση: Όταν το συγκεκριμένο (N) Bit είναι 1 τότε η πράξη μας δίνει την τιμή  $2^N$

# Turn off a bit

Όταν θέλουμε να κάμουμε ενα συγκεκριμένο bit 0 τότε χρησιμοποιούμε NOT και AND.

x	34	00100010
y	1<<1	00000010
~y	~(1<<1)	11111101
x=x&(~y)	42	00100000

# Switch a bit

Όταν θέλουμε να αντιστρέψουμε την τιμή ενός bit τότε χρησιμοποιούμε XOR.

x	40	00101000
y	$1 \ll 2$	00000100
$x = x \oplus y$	44	00101100

x	40	00101000
y	$1 \ll 3$	00001000
$x = x \oplus y$	32	00100000

# Εύρεση τελευταίου bit

Όταν θέλουμε να βρούμε την τιμή του τελευταίου bit χρησιμοποιούμε AND με τον αρνητικό του.

x	40	000...00010 <b>1</b> 000
x	-40	111...11101 <b>1</b> 000
t=(x & -x)	8	000...00000 <b>1</b> 000

# Turn on n bits

Όταν θέλουμε να κρατικοποιήσουμε τα πρώτα N bits με 1 τότε το κάνουμε με την βοήθεια του Shift.

x	$1 \ll 5$	00100000
$x = x - 1$	$(1 \ll 5) - 1$	00011111



There are 10 types of people in the world:  
Those who understand binary  
and  
Those who don't

