papergrid
Date: / /

1. 0b11010111

To Swap big Endian to little Endian.

It is a 8 bit Value

0b00000000

big Endian = 0b11010111

little = 0b11010111

Left most value = 0b000000FF

Left middle val = 0b0000FF00

((0x00000 0ff & Value) << 24)

(0x0000

<<

=

0 != ((0x000000ff & Value) << 24

0x00000011

0x11223344

0x11223300

0x11223301 = (00112200 44)

papergrid
Date: / /

AND			OR		
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

0x000000FF

01000001

0x59656969

00111011

01000101

01000104

0x59697969

0x69797959

0x12345678

0x87654321

87 - 01010111

78 ⇒ 01001110

56 ⇒ 00111000

34 ⇒ 00100010

00001100

papergrid

Date: / /

y = 00001110

y << 4.

y = 1110 0000

4 E

0100 1110

y = 0100 0000

α = 4E.

output = E4

y = 0000 0100

4E & 0F

0000 1111

0100 1110
0000 1111
0000 1110

β = 0000 1110

11100000
00000100
11000000
E4

4E & F0

y = 0100 0000

0100 1110
1111 0000
0100 0000

~~8×4=32~~
1111

papergrid
Date: / /

A = 0x12345678

A = 0000 1100 0010 0010 0011 1000 0100 1110
→ 0x1234 5678

0111 0100 0001 1100 0100 0100 0011 0000
→ 0x8765 4321

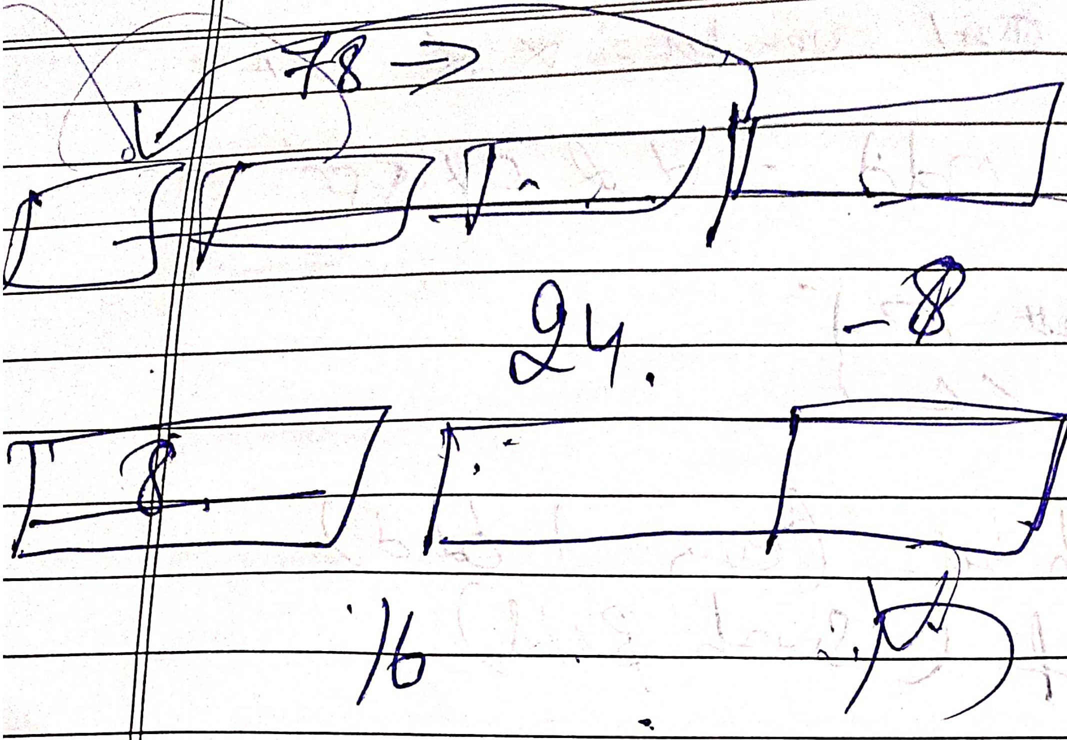
~~78~~ 87

0100
1110

0101 0111 & 0000 1111

0101 0111
0000 1111
0000 0111

0101 0111
1111 0000
0101 0000



23 = 00010111 &

0XAA = 10101010

00000010

even

First we take

01010101

vice versa

odd <<

Then the Program

#include<stdio.h>

unsigned int x; even, odd, sw

printf

papergrid

Date: / /

unsigned int even bit = x & 0xAA

odd = x & 0x55

even bit >> 1
odd << 1

Swap bits (even | odd)
printf (swap Bits)

---

x

```
    *
   ***
  *****
 *******
*********
```

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

↓

Rows → Input

i (1) (1)
1 → 1 → 1   Space 3 1 4

2 → 3   2 2 4

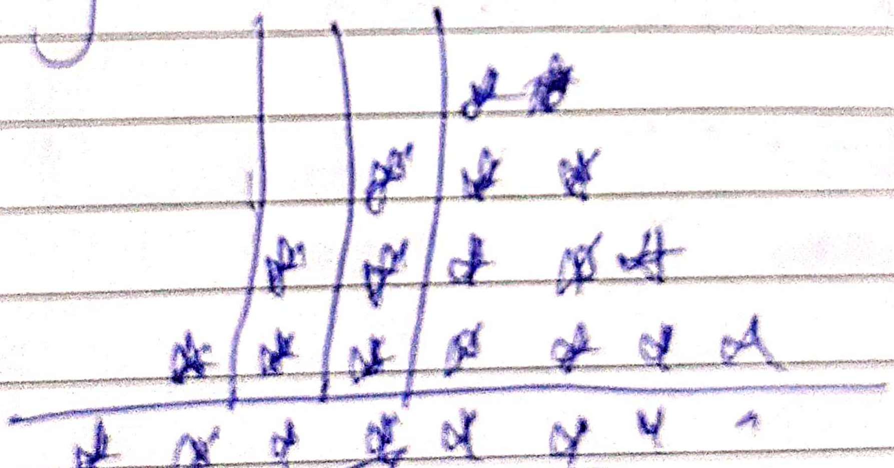
3 → 5   1 3 4

(2i - 1)   0 4 4

4 → 7

papergrid

```
for (Space=1; Space<=row-i; Space++)
For(i=1; i<=rows; i++)
{
    for (j=1; j<=2*i-1; j++)
    {
        Print("*");
    }
    Print("\n");
}
```



Total	Row	Star	Space → n-1-1
n	i	j	
5	0	1	4
5	1	3	3
5	2	5	2
5	3	7	1

5-0-1=4

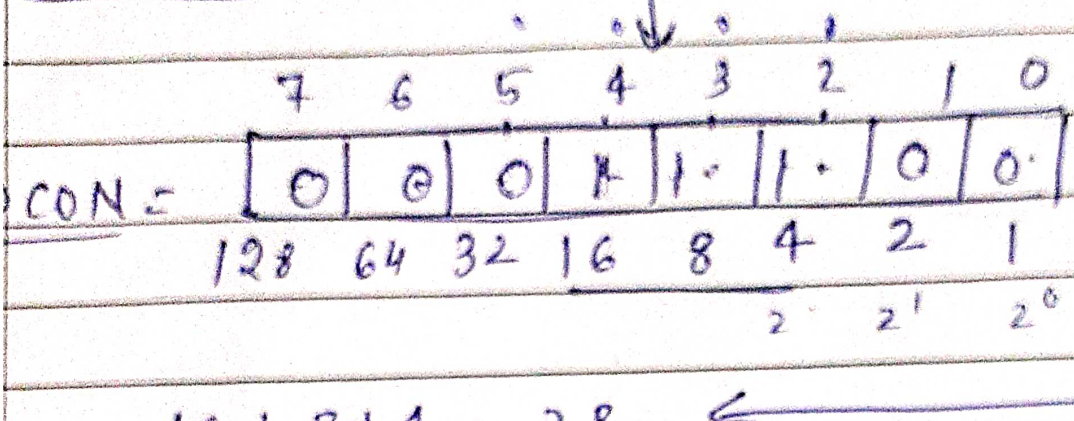
2j=1*2+1

papergrid

Date: / /

ADCON = 0X07

CHCON



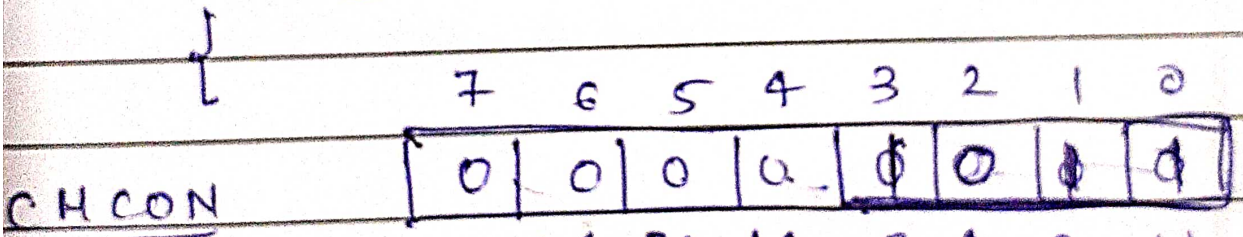
16 + 8 + 4 = 28 dec

if (~~ADCON~~ 28 & ADCON == 28)

ADCON: 1.0 1 0 1 . 1 0 1

4 0 0 0 1 1 1 0 0

0 0 0 0 1 1 1 0 0 = 28



COMCON = COMCON & 0XF0

=

0 0 0 0 | 1 0 1 1

& a a a a | 0 0 0 0

0 0 0 0 0 0 0 0

printf ("%x", COMCON)