PBM, PGM, and PPM

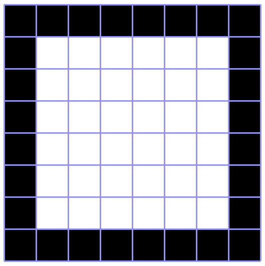
- ***PBM*** (Portable ***Bit*** Map), ***PGM*** (Portable ***Gray*** Map), and ***PPM*** (Portable ***Pixel*** Map) are three simple bitmap image formats
- In ***PBM***, ***PGM***, or ***PPM*** image format
  - □ An image consists of image header followed by image data
  - □ Headers are always written in ASCII
  - □ Header values are separated by white space, i.e., ***Tab***, ***<CR>***, or ***Space***
  - □ Characters coming just after a "***#***" character to the next ***<CR>*** are considered a comment and ignored
  - □ Image data comes immediately after the last header field
  - □ Image data is a series of values describing image pixels in *bitmap format*
  - □ The bitmap pixels
    - start at the top-left corner of the image
    - proceed from left to right, and from top to bottom in a *raster scan fashion*
  - □ Image data is either
    - ***ASCII:*** values are separated by white space, or
    - ***Raw:*** values are stored without any white space between them

PBM

- A ***PBM*** header consists of
  - □ ***Magic value*** (a.k.a. ***file identifier value***) the first two bytes of the header
    - P1 for *ASCII* pixel values
    - P4 for *raw* pixel values
  - □ ***Image width***: ASCII decimal value
  - □ ***Image height***: ASCII decimal value

Example

```
P1
# A PBM example
8 8 # width and height
1 1 1 1  1 1 1 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 1 1 1   1 1 1 1 # This is the end of the data
```



PBM

- Note that, in ***PBM*** format:
  - □ "*1*" means *black*
  - □ "*0*" means *white*

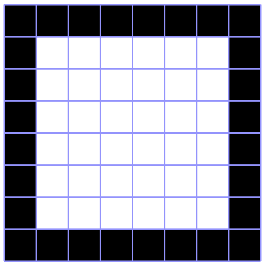
  This is **not** the case in ***PGM*** and ***PPM***

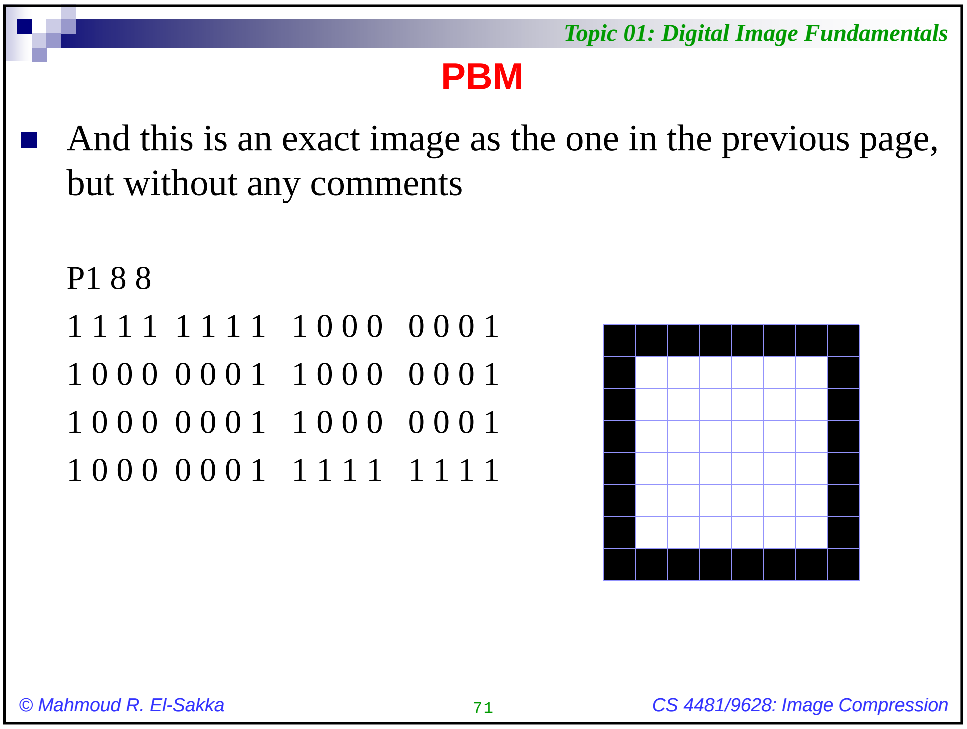
- Any number, other than "*0*" or "*1*", in the pixel values area is *not acceptable* (must be binary values only)

PBM

- This is an exact image as the one in the previous page, but with more comments

```
P1 # A PBM example
8 #width
8 # height
#
1 1 1 1  1 1 1 1   1 0 0 0   0 0 0 1
# The following is the 3rd and 4th rows
1 0 0 0  0 0 0 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 0 0 0   0 0 0 1
# The following is part of the 5th row and the 6th row
1 0 0 0  0 0 0 1   1 1 1 1   1 1 1 1
```

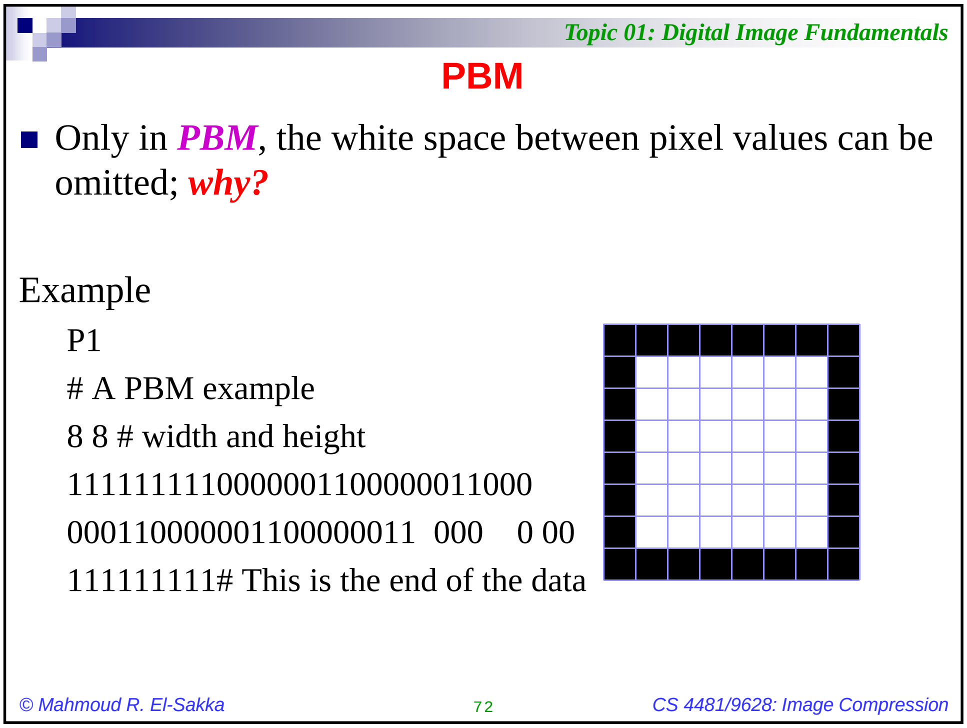




PBM

- And this is an exact image as the one in the previous page, but without any comments

```
P1 8 8
1 1 1 1  1 1 1 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 0 0 0   0 0 0 1
1 0 0 0  0 0 0 1   1 1 1 1   1 1 1 1
```



PBM

- Only in ***PBM***, the white space between pixel values can be omitted; ***why?***

Example

```
P1
# A PBM example
8 8 # width and height
1111111110000001100000011000
0001100000011000000011  000    0 00
111111111# This is the end of the data
```

PBM

- In case of P4
  - □ each image row ***must be*** stored as raw bits in
    - *eight pixels per byte* (*MSB first*)

    How about *big-endian* and *little-endian*?

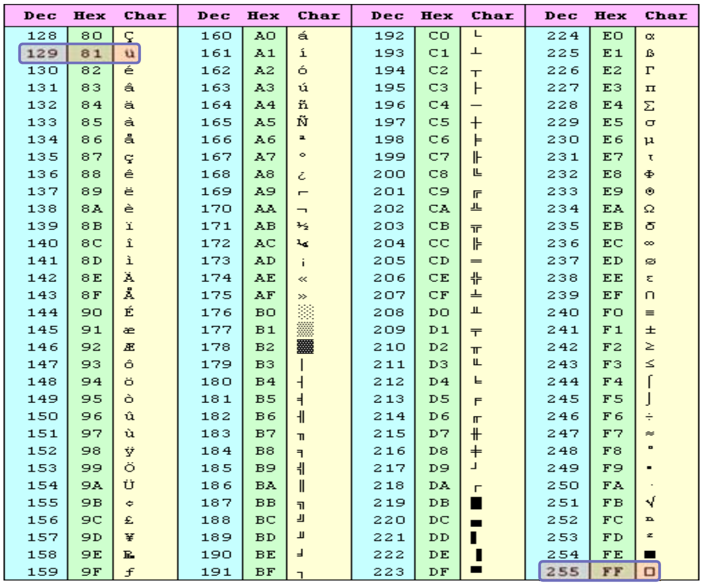
  - □ no two rows share same byte
    (the rest of the last byte of each row to be filled by zeros, if needed)
  - □ no white space or comment is allowed in the data section, after all it is no longer ASCII

Describe the data in the previous binary image, after converting it to a raw PBM format (P4)

ASCII Table

0	NUL	1	SOH	2	STX	3	ETX	4	EOT	5	ENQ	6	ACK	7	BEL
8	BS	9	HT	10	NL	11	VT	12	NP	13	CR	14	SO	15	SI
16	DLE	17	DC1	18	DC2	19	DC3	20	DC4	21	NAK	22	SYN	23	ETB
24	CAN	25	EM	26	SUB	27	ESC	28	FS	29	GS	30	RS	31	US
32	SP	33	!	34	"	35	#	36	$	37	%	38	&	39	‘
40	(	41	)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[	92	\	93	]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	DEL

ASCII Table



Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

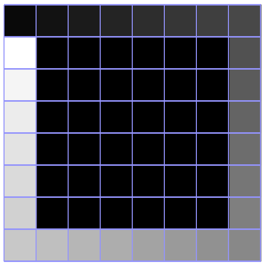
PGM

- A ***PGM*** header consists of
  - □ ***Magic value*** (the first two bytes of the header)
    - P2 for *ASCII* pixel values
    - P5 for *raw* pixel values
  - □ ***Image width***: ASCII decimal value
  - □ ***Image height***: ASCII decimal value
  - □ ***Maximum gray value***: ASCII decimal value
    - Must be less than 65536 and more than zero
    - Will affect the number of bytes per pixel in raw format; ***how?***

PGM

- Example

```
P2 8 8 28 # P2, width, height, maximum gray value
 1  2  3  4   5  6  7   8    28   0   0   0    0   0   0   9
27  0  0  0   0  0  0  10    26   0   0   0    0   0   0 11
25  0  0  0   0  0  0  12    24   0   0   0    0   0   0 13
23  0  0  0   0  0  0  14    22  21 20 19   18 17 16 15
```



- Note that in ***PGM*** format
  - □ “*0*” means *black*,
This is **not** the case in ***PBM***
  - □ *maximum gray value* means *white*
- Any value greater than *maximum gray value* is *likely interpreted* as “*0*”, depends on the viewer
- If P5 is selected, image data must be stored as raw bits in one byte (or two bytes) per pixel

Describe the data in the this gray image, after converting it to a raw PGM format (P5)

ASCII Table

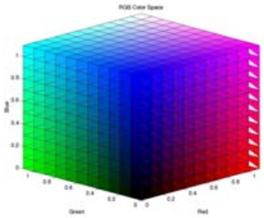
0	NUL	1	SOH	2	STX	3	ETX	4	EOT	5	ENQ	6	ACK	7	BEL
8	BS	9	HT	10	NL	11	VT	12	NP	13	CR	14	SO	15	SI
16	DLE	17	DC1	18	DC2	19	DC3	20	DC4	21	NAK	22	SYN	23	ETB
24	CAN	25	EM	26	SUB	27	ESC	28	FS	29	GS	30	RS	31	US
32	SP	33	!	34	"	35	#	36	$	37	%	38	&	39	‘
40	(	41	)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[	92	\	93	]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	DEL

PPM

- A ***PPM*** header consists of
  - □ ***Magic value*** (the first two bytes of the header)
    - P3 for *ASCII* pixel values
    - P6 for *raw* pixel values
  - □ ***Image width***: ASCII decimal value
  - □ ***Image height***: ASCII decimal value
  - □ ***Maximum color value***: ASCII decimal value
    - Must be less than 65536 and more than zero
    - Will affect the number of bytes per pixel in raw format; ***how?***

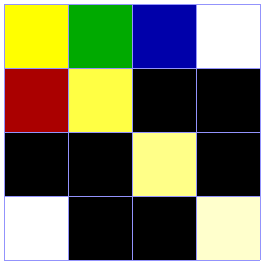
PPM

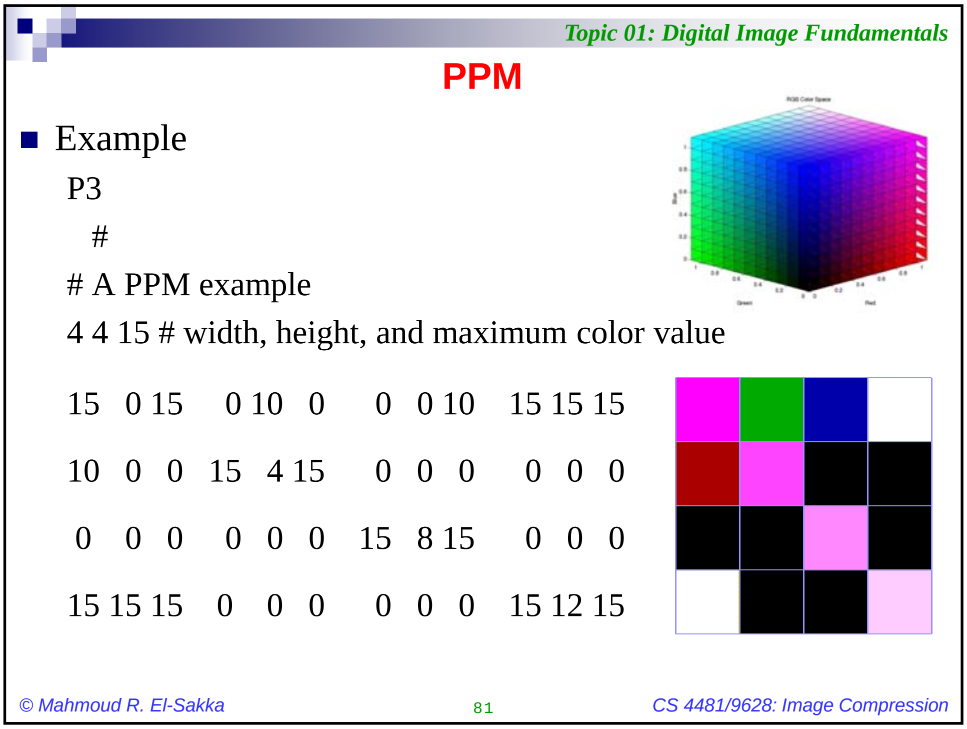
- Example



```
P3
 #
# A PPM example
4 4 15 # width, height, and maximum color value

15 15  0    0 10  0    0  0 10   15 15 15
10  0  0   15 15  4    0  0  0    0  0  0
 0  0  0    0  0  0   15 15  8    0  0  0
15 15 15    0  0  0    0  0  0   15 15 12
```



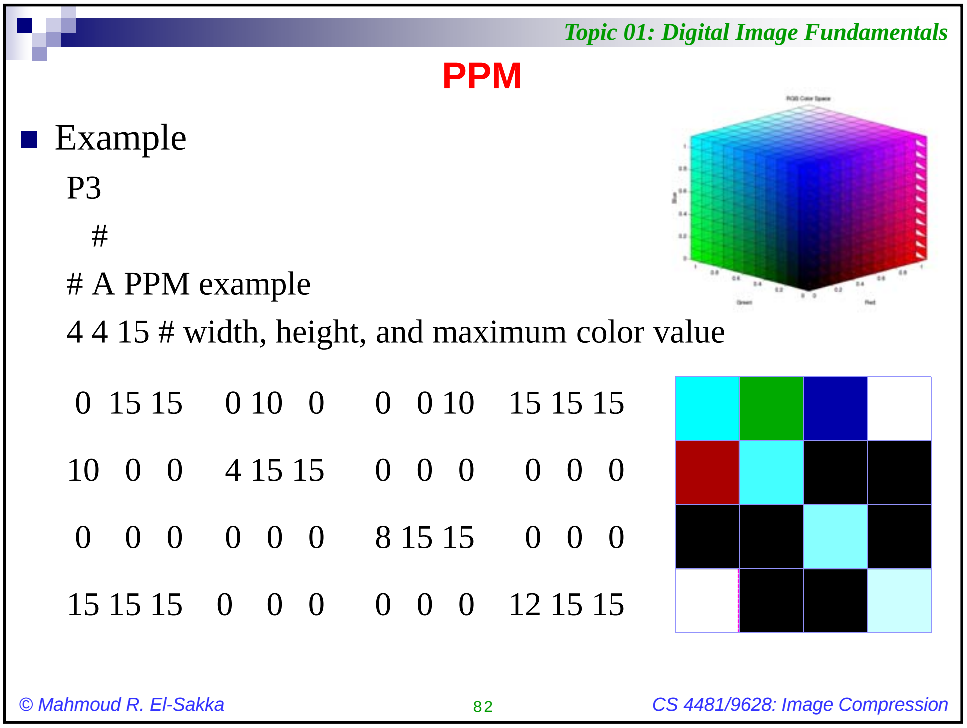


PPM

- Example

```
P3
  #
# A PPM example
4 4 15 # width, height, and maximum color value

15   0 15     0 10   0      0   0 10    15 15 15
10   0   0   15   4 15      0   0   0     0   0   0
 0   0   0     0   0   0   15   8 15     0   0   0
15 15 15   0     0   0      0   0   0    15 12 15
```

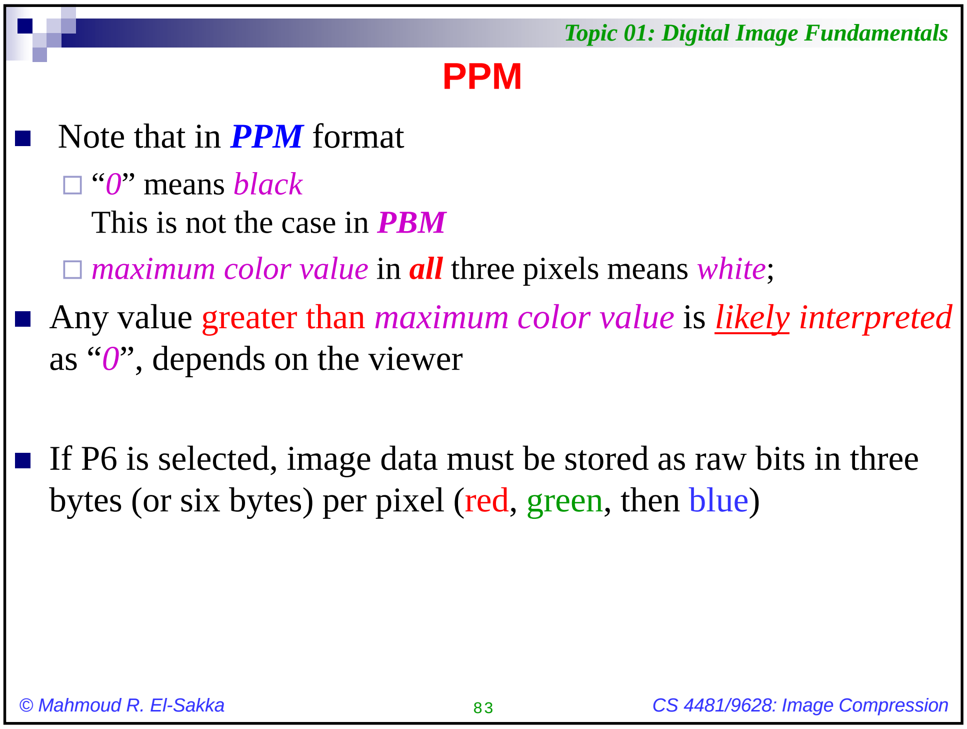


PPM

- Example

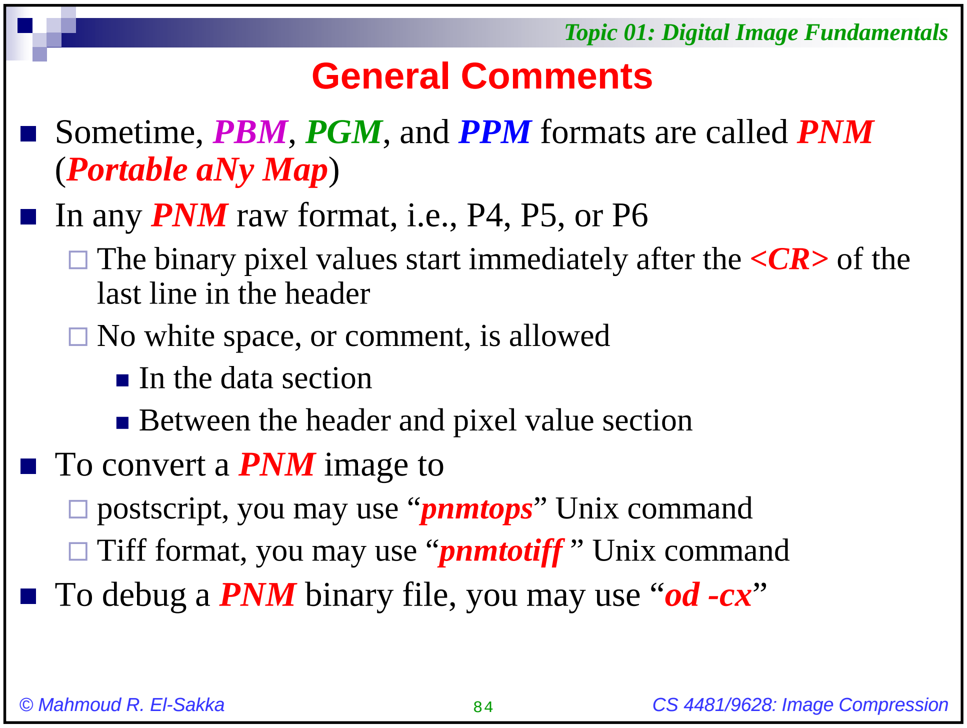
```
P3
  #
# A PPM example
4 4 15 # width, height, and maximum color value

 0  15 15     0 10   0      0   0 10    15 15 15
10   0   0     4 15 15      0   0   0     0   0   0
 0   0   0     0   0   0    8 15 15     0   0   0
15 15 15   0     0   0      0   0   0    12 15 15
```



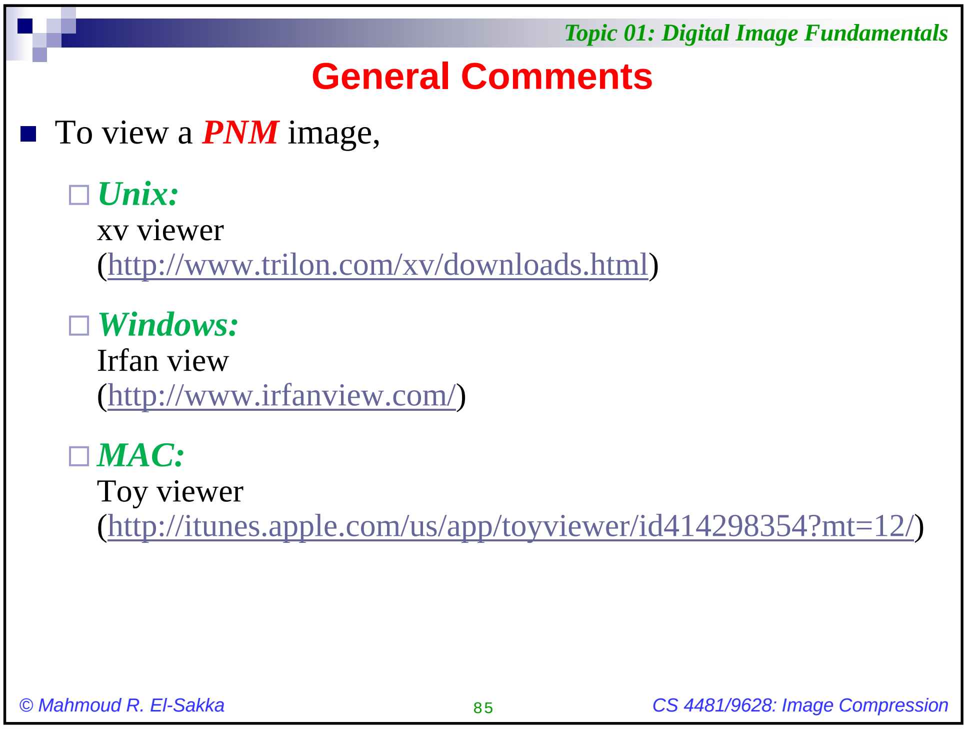
PPM

- Note that in ***PPM*** format
  - “*0*” means *black*
    This is not the case in ***PBM***
  - *maximum color value* in ***all*** three pixels means *white*;
- Any value greater than *maximum color value* is *likely interpreted* as “*0*”, depends on the viewer
- If P6 is selected, image data must be stored as raw bits in three bytes (or six bytes) per pixel (red, green, then blue)



General Comments

- Sometime, ***PBM***, ***PGM***, and ***PPM*** formats are called ***PNM*** (***Portable aNy Map***)
- In any ***PNM*** raw format, i.e., P4, P5, or P6
  - The binary pixel values start immediately after the ***<CR>*** of the last line in the header
  - No white space, or comment, is allowed
    - In the data section
    - Between the header and pixel value section
- To convert a ***PNM*** image to
  - postscript, you may use “***pnmtops***” Unix command
  - Tiff format, you may use “***pnmtotiff*** ” Unix command
- To debug a ***PNM*** binary file, you may use “***od -cx***”



General Comments

- To view a ***PNM*** image,
  - ***Unix:***
    xv viewer
    (http://www.trilon.com/xv/downloads.html)
  - ***Windows:***
    Irfan view
    (http://www.irfanview.com/)
  - ***MAC:***
    Toy viewer
    (http://itunes.apple.com/us/app/toyviewer/id414298354?mt=12/)