SER 334 A Session

SI Session

Sunday, January 21st 2024

7:00 pm - 8:00 pm MST

Agenda

- Macros
- Function Parameters
- BMP File Format
  - Reading Contents
  - Headers

SI Session Expectations

Thanks for coming to the **SER 334** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
  - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features

Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter's questions.
- Annotations are encouraged

SER 334
Macros

Very helpful **BUT** we need to be *careful*

```c
#define username "Katie"
#define PI 3.14
#define square(s) s*s

int main(){

    printf( format: "%s\n", username);
    printf( format: "%f\n", PI);
    printf( format: "%d\n", square(3));
    printf( format: "%d\n", square(3+7));

    return 0;
}
```

```
Katie
3.140000
9
31   ?
```

How can we solve this?

SER 334
Parameters

```
void funcOne(int n) {
```

```
void funcTwo(int *n) {
```

Both functions will perform the same task

BUT there is a big difference in *implementation*

SER 334
Parameters

```c
int r = 5;
printf( format: "Before Func One\n");
printf( format: "r: %d\n", r);
printf( format: "&r: %d\n", &r);
funcOne( n: r);
printf( format: "r: %d\n", r);
printf( format: "&r: %d\n", &r);
```

Standard Pass-By-Value, same as we have before.

```c
void funcOne(int n) {
    printf( format: "FuncOne received: %d\n", n);
    printf( format: "FuncOne received &n: %d\n", &n);
    int x = n;
    n = 0;
    printf( format: "\tWithin FuncOne\n\tx: %d\n\tn: %d\n",x , n);
}
```

SER 334
Parameters

```c
int r = 5;
printf( format: "Before Func One\n");
printf( format: "r: %d\n", r);
printf( format: "&r: %d\n", &r);
funcOne( n: r);
printf( format: "r: %d\n", r);
printf( format: "&r: %d\n", &r);
```

```
Before Func One
r: 5
&r: 2084567436
FuncOne received: 5
FuncOne received &n:
        Within FuncOne
        x: 5
        n: 0
r:
&r: 2084567436
```

Standard Pass-By-Value, same as we have before.

```c
void funcOne(int n) {
    printf( format: "FuncOne received: %d\n", n);
    printf( format: "FuncOne received &n: %d\n", &n);
    int x = n;
    n = 0;
    printf( format: "\tWithin FuncOne\n\tx: %d\n\tn: %d\n",x , n);
}
```

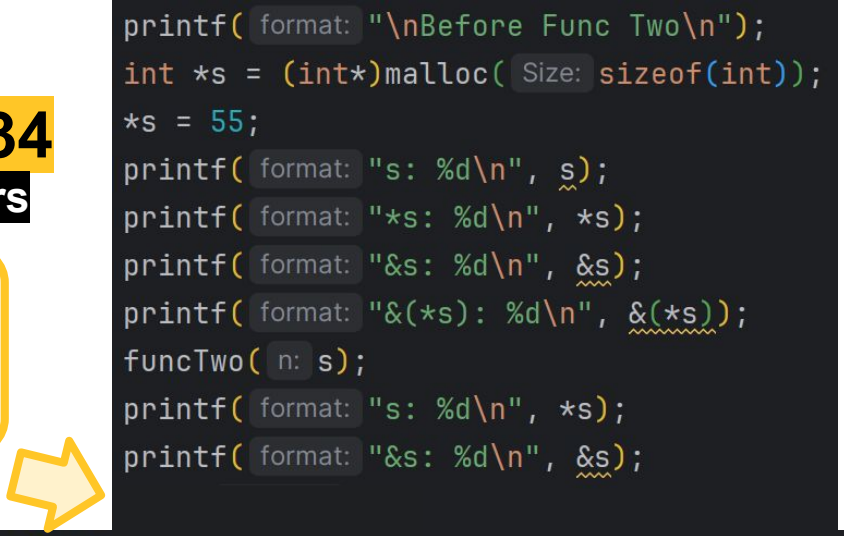
Will this be the same address as r?

SER 334

Parameters

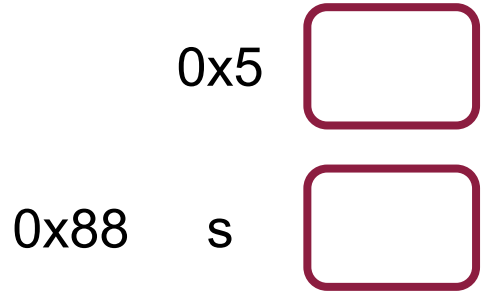
Missing something here?

```
printf( format: "\nBefore Func Two\n");
int *s = (int*)malloc( Size: sizeof(int));
*s = 55;
printf( format: "s: %d\n", s);
printf( format: "*s: %d\n", *s);
printf( format: "&s: %d\n", &s);
printf( format: "&(*s): %d\n", &(*s));
funcTwo( n: s);
printf( format: "s: %d\n", *s);
printf( format: "&s: %d\n", &s);
```



```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Pass-By-Value, **using a pointer**.



SER 334

Parameters

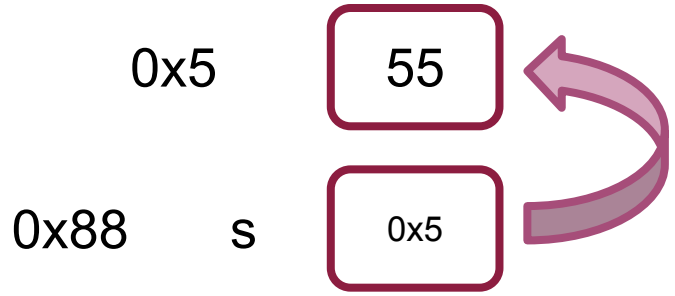
```
printf( format: "\nBefore Func Two\n");
int *s = (int*)malloc( Size: sizeof(int));
*s = 55;
printf( format: "s: %d\n", s);
printf( format: "*s: %d\n", *s);
printf( format: "&s: %d\n", &s);
printf( format: "&(*s): %d\n", &(*s));
funcTwo( n: s);
printf( format: "s: %d\n", *s);
printf( format: "&s: %d\n", &s);
free( Memory: s);
```

```
Before Func Two
s: -871623632
*s: 55
&s: 2084567424
&(*s): -871623632
```

```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Pass-By-Value, **using a pointer**.

SER 334
Parameters



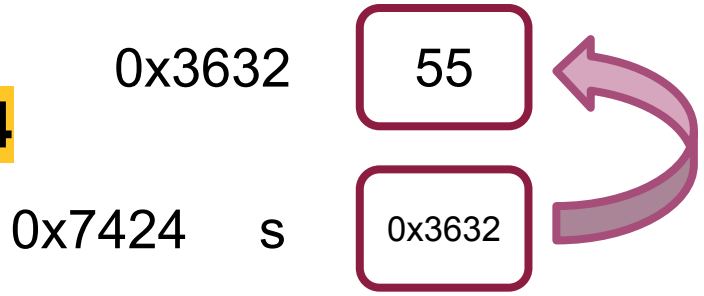

```
Before Func Two
s: -871623632
*s: 55
&s: 2084567424
&(*s): -871623632
```

```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Pass-By-Value, **using a pointer**.

SER 334

Parameters



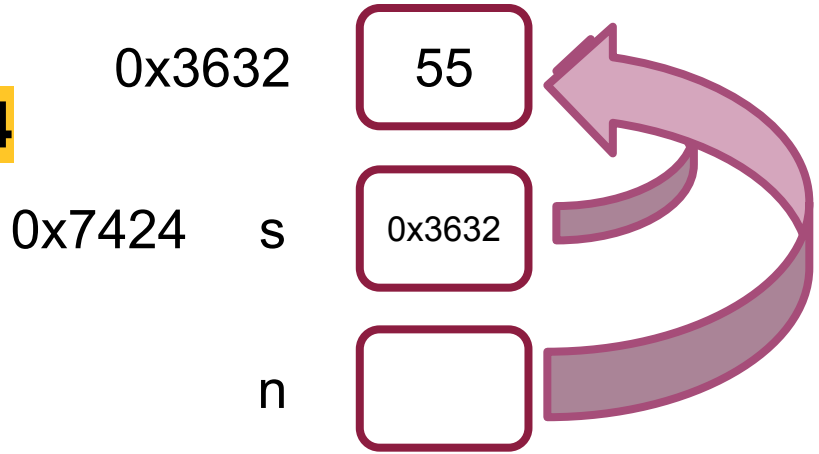

```
Before Func Two
s: -871623632
*s: 55
&s: 2084567424
&(*s): -871623632
```

```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Pass-By-Value, **using a pointer**.

SER 334

Parameters




```
Before Func Two
s: -871623632
*s: 55
&s: 2084567424
&(*s): -871623632
FuncTwo received     n: -871623632
FuncTwo received    *n: 55
FuncTwo received    &n: 2084567376
FuncTwo received &(*n): -871623632
        Within FuncTwo
        x: 55
        n: 0
s: 0
&s: 2084567424
```

```c
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Pass-By-Value, **using a pointer**.

SER 334
Parameters

Sent the ***location*** of the variable!

```
printf( format: "\nBefore Func Two Run Two\n");
int t = 55;
printf( format: "t: %d\n", t);
printf( format: "&t: %d\n", &t);
funcTwo( n: &t);
printf( format: "t: %d\n", t);
printf( format: "&t: %d\n", &t);
```

```
Before Func Two Run Two
t: 55
&t: 2084567420
```

Also notice that we are using a normal int and not a pointer!

```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Same Function, but **Pass-By-Reference**

SER 334
Parameters

Sent the ***location*** of the variable!

```
printf( format: "\nBefore Func Two Run Two\n");
int t = 55;
printf( format: "t: %d\n", t);
printf( format: "&t: %d\n", &t);
funcTwo( n: &t);
printf( format: "t: %d\n", t);
printf( format: "&t: %d\n", &t);
```

```
Before Func Two Run Two
t: 55
&t: 2084567420
```

```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

SER 334
Parameters

Sent the ***location*** of the variable!

```
printf( format: "\nBefore Func Two Run Two\n");
int t = 55;
printf( format: "t: %d\n", t);
printf( format: "&t: %d\n", &t);
funcTwo( n: &t);
printf( format: "t: %d\n", t);
printf( format: "&t: %d\n", &t);
```

```
Before Func Two Run Two
t: 55
&t: 2084567420
FuncTwo received     n: 2084567420
FuncTwo received    *n: 55
FuncTwo received    &n: 2084567376
FuncTwo received &(*n): 2084567420
        Within FuncTwo
        x: 55
        n: 0
t: 0
&t: 2084567420
```

```
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",
}
```

0x7420	t	55
	n	

SER 334

Parameters

Sent the ***location*** of variable!

What about sending *the location* of a pointer? Any ideas?

```
printf( format: "\nBefore Func Two Run Two\n");
int t = 55;
printf( format: "t: %d\n", t);
```

```
Before Fur
t: 55
&t: 2084
FuncTwo
FuncTwo
FuncTwo r
FuncTwo rece
Within
x: 55
n: 0
t: 0
&t: 2084567420
```

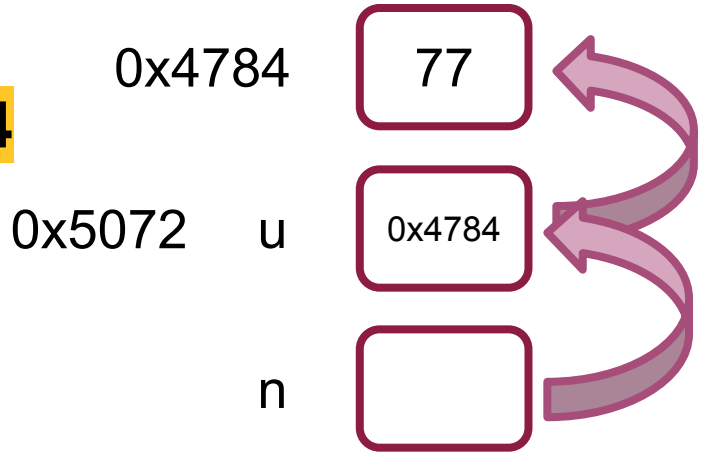
```
void func
    print
    print
    print
    print
    int x
    *n =
    print
}
```

```
printf( format: "\nBefore Func Two Run Three\n");
int *u = (int*)malloc( Size: sizeof(int));
*u = 77;
printf( format: "u: %d\n", u);
printf( format: "*u: %d\n", *u);
printf( format: "&u: %d\n", &u);
printf( format: "&(*u): %d\n", &(*u));
funcTwo( n: &u);
printf( format: "u: %d\n", u);
printf( format: "ut: %d\n", &u);
free( Memory: u);
```

0x7420	t	55
0x7376	n	0x7420

SER 334

Parameters




```
Before Func Two Run Three
u: -1750264784
*u: 77
&u: -77595072
&(*u): -1750264784
FuncTwo received     n: -77595072
FuncTwo received    *n: -1750264784
FuncTwo received    &n: -77595104
FuncTwo received &(*n): -77595072
        Within FuncTwo
        x: -1750264784
        n: 0
u: 0
ut: -77595072
```

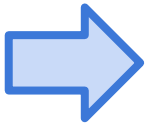
```c
void funcTwo(int *n) {
    printf( format: "FuncTwo received     n: %d\n", n);
    printf( format: "FuncTwo received    *n: %d\n", *n);
    printf( format: "FuncTwo received    &n: %d\n", &n);
    printf( format: "FuncTwo received &(*n): %d\n", &(*n));
    int x = *n;
    *n = 0;
    printf( format: "\tWithin FuncTwo\n\tx: %d\n\tn: %d\n",x , *n);
}
```

Pointer to a pointer - we don't have the value!

SER 334
Parameters

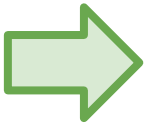
Summary

```
void funcOne(int n) {
```



Makes a local copy

```
void funcTwo(int *n) {
```

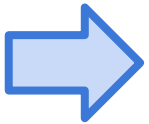


Makes a pointer

SER 334
Parameters

Summary

```
void funcOne(int n) {
```



Makes a local copy

One pointer exists outside of funcTwo, one pointer exists within funcTwo

```
void funcTwo(int *n) {
```

Pointer

Integer

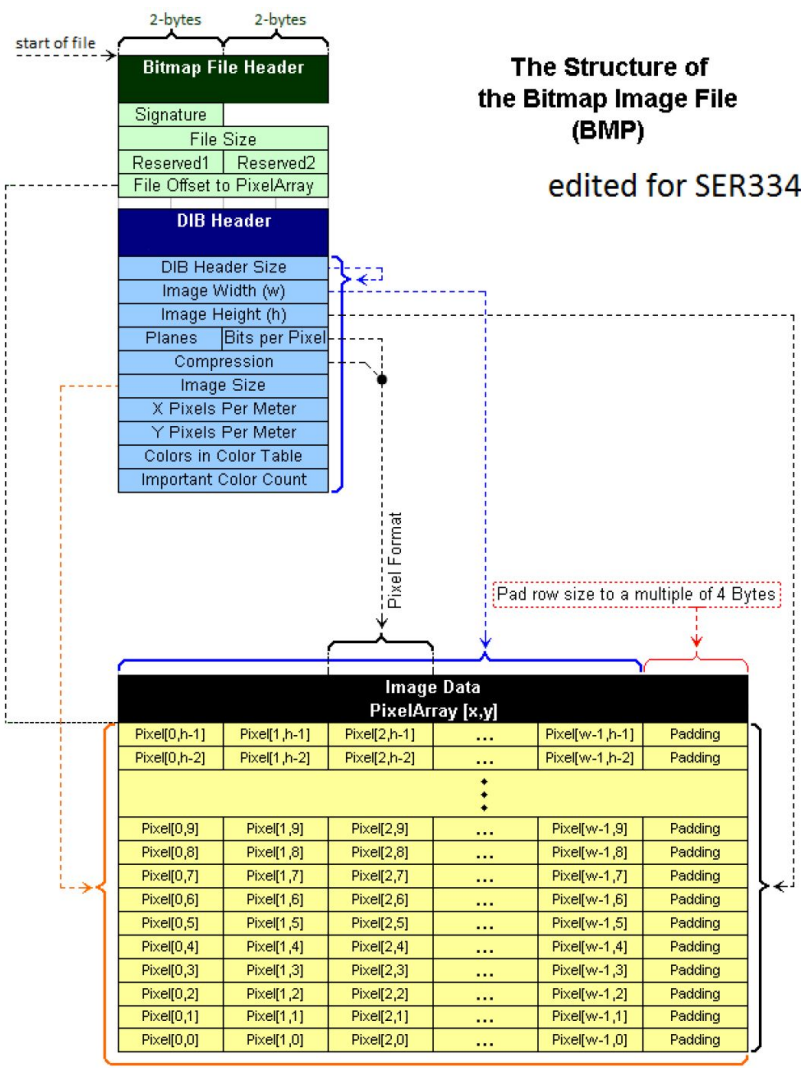
Integer exists outside of funcTwo, the pointer exists within funcTwo

Makes a new pointer, that is pointing to the same address as the parameter

??

Makes a new pointer, that is pointing to the same address as the parameter

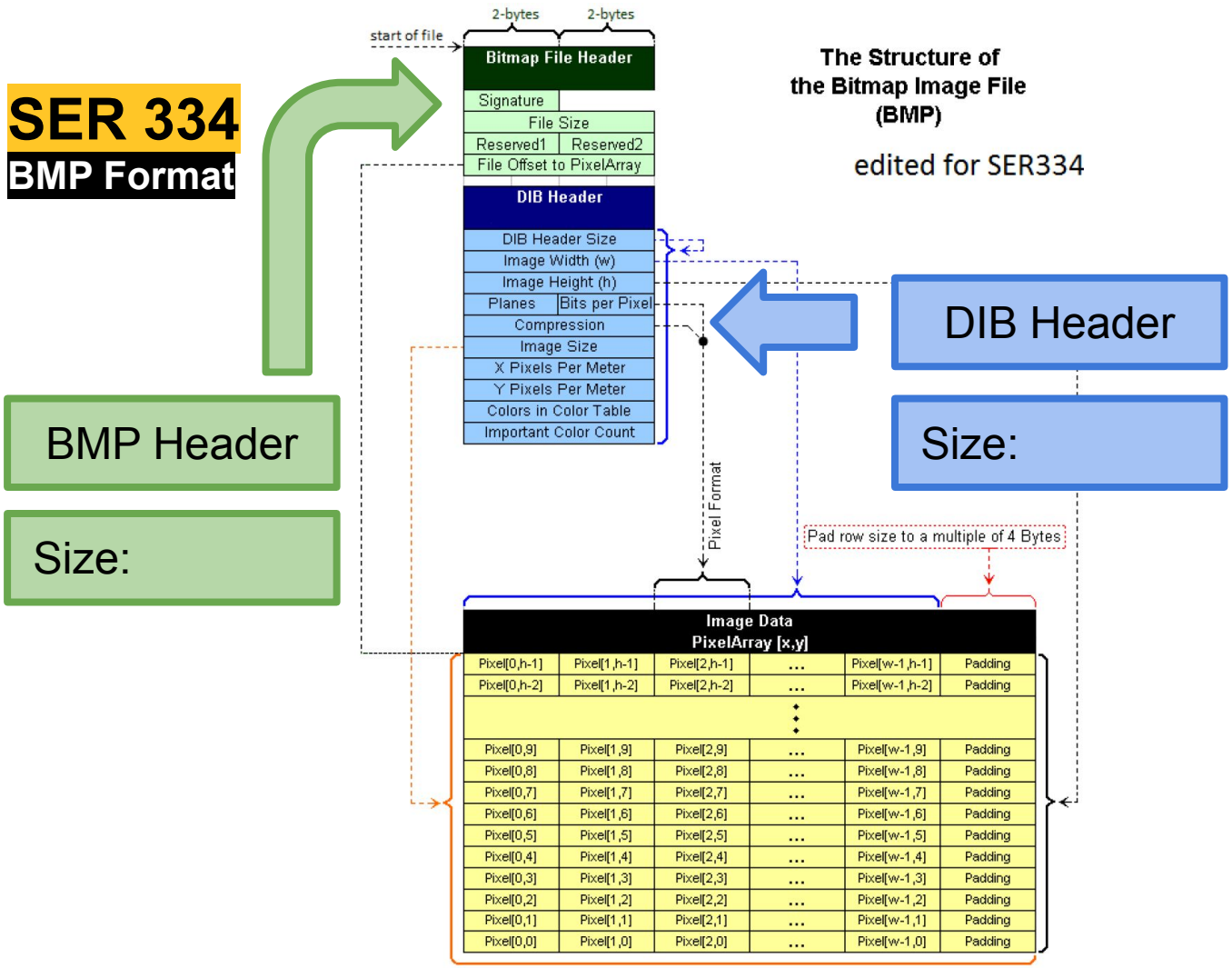
SER 334
BMP Format



BMP Header Starter Code

SER 334
BMP Format

BMP Header Starter Code

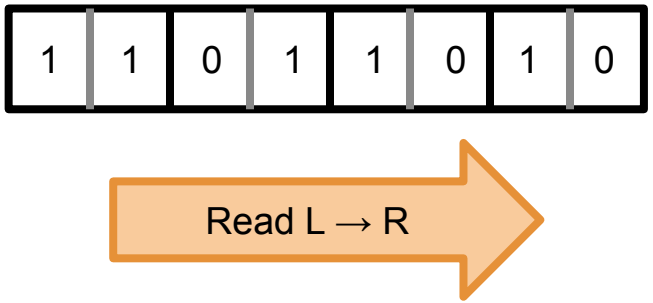


SER 334

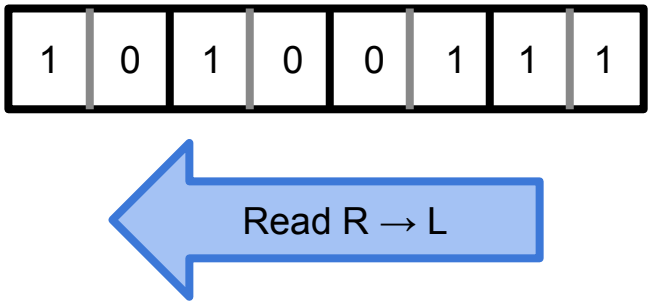
BMP File Format

Remember to account for Endianness!

Big Endian **HEX**



Little Endian **HEX**



Decimal: 285, 282, 320

Binary: 0001 0001 0000 0001 0001 0000 0001 0000

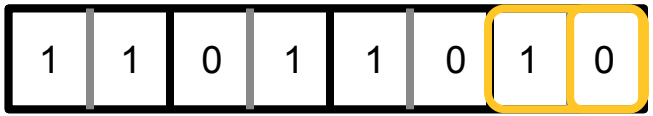
SER 334

BMP File Format

Remember to account for Endianness!

What is the size of this section?

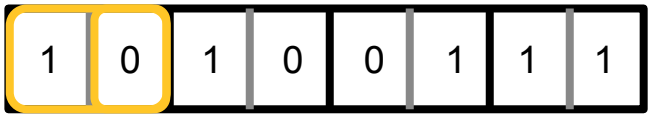
Big Endian **HEX**



Read L → R

1 byte

Little Endian **HEX**



Read R → L

Decimal: 285, 282, 320

Binary: 0001 0001 0000 0001 0001 0000 0001 0000

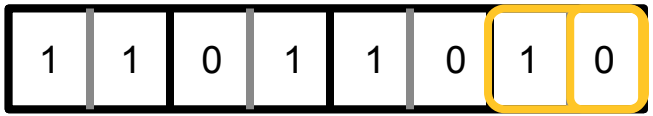
SER 334

BMP File Format

Remember to account for Endianness!

What is the size of this section?

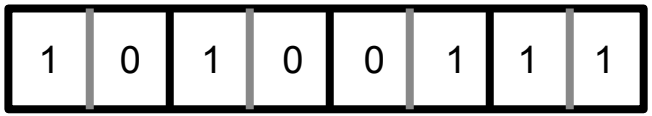
Big Endian **HEX**



Read L → R

1 byte

Little Endian **HEX**



Read R → L

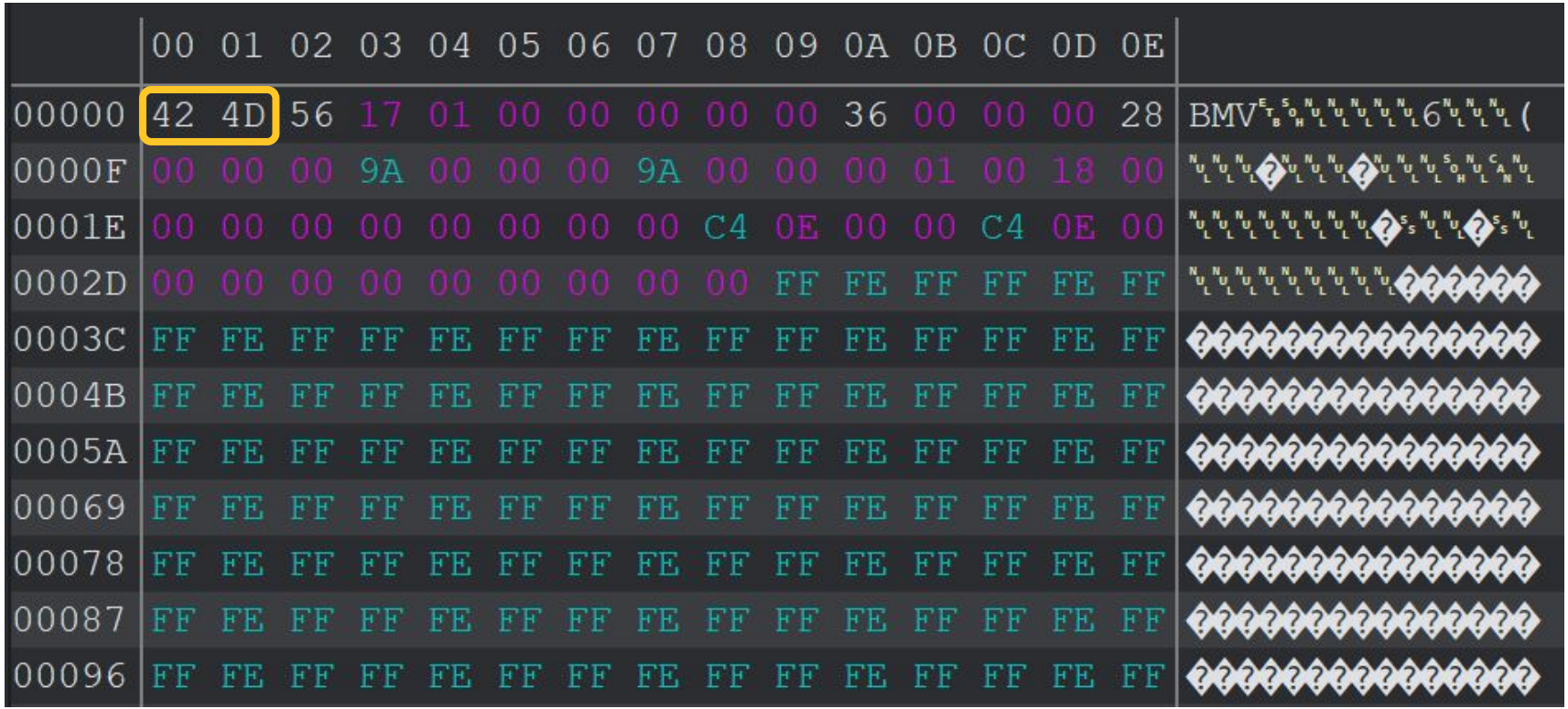
Decimal: 285, 282, 320

Binary: 0001 0001 0000 0001 0001 0000 0001 0000

SER 334

BMP File Format

What is this piece?

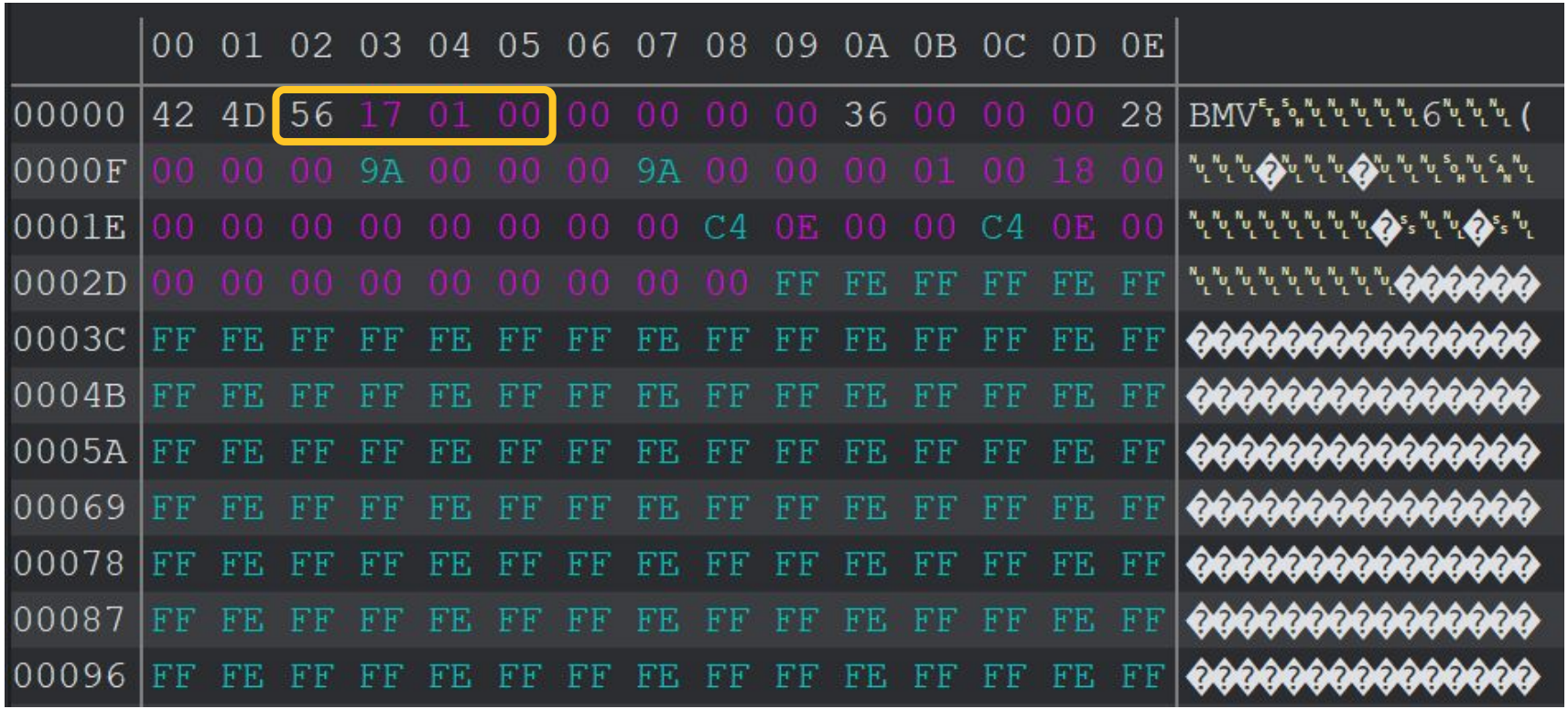


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

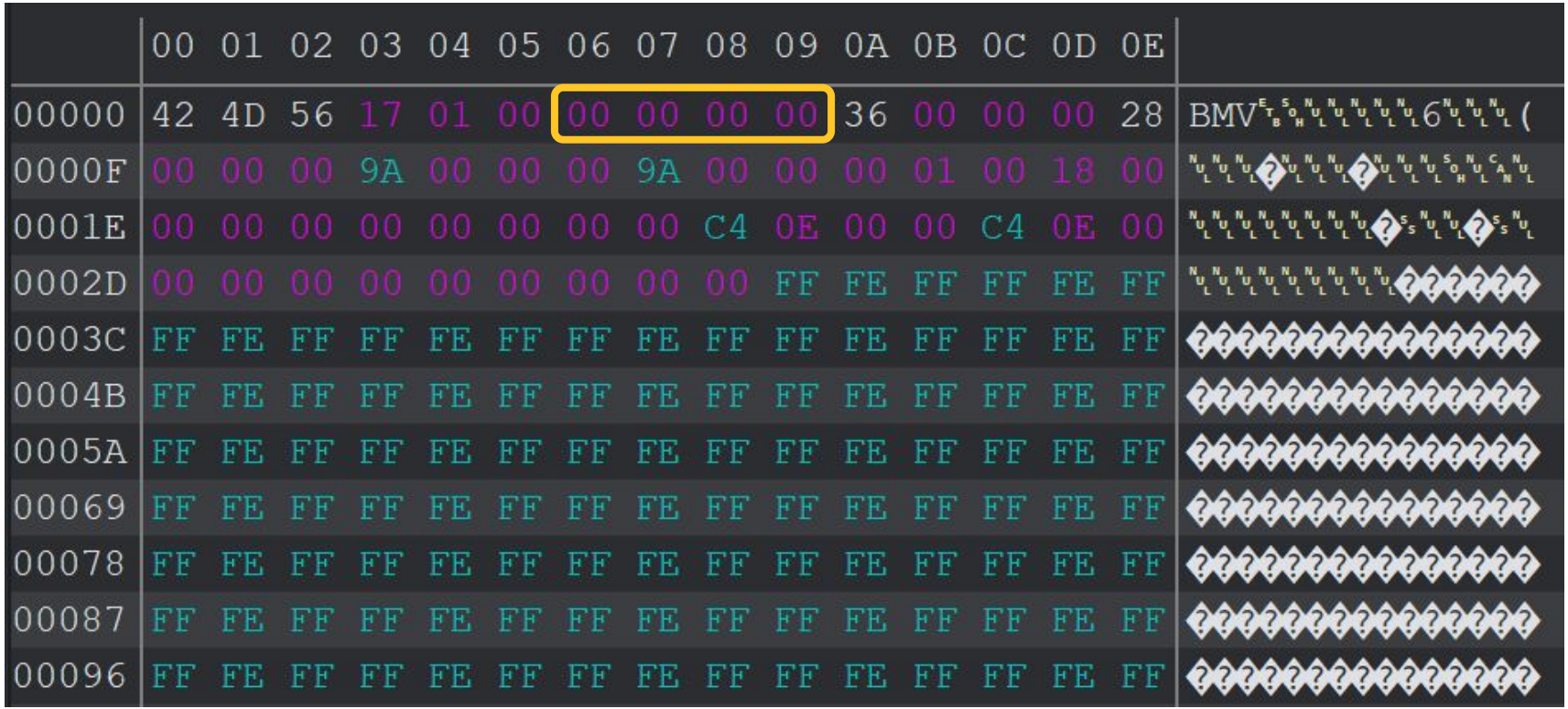


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

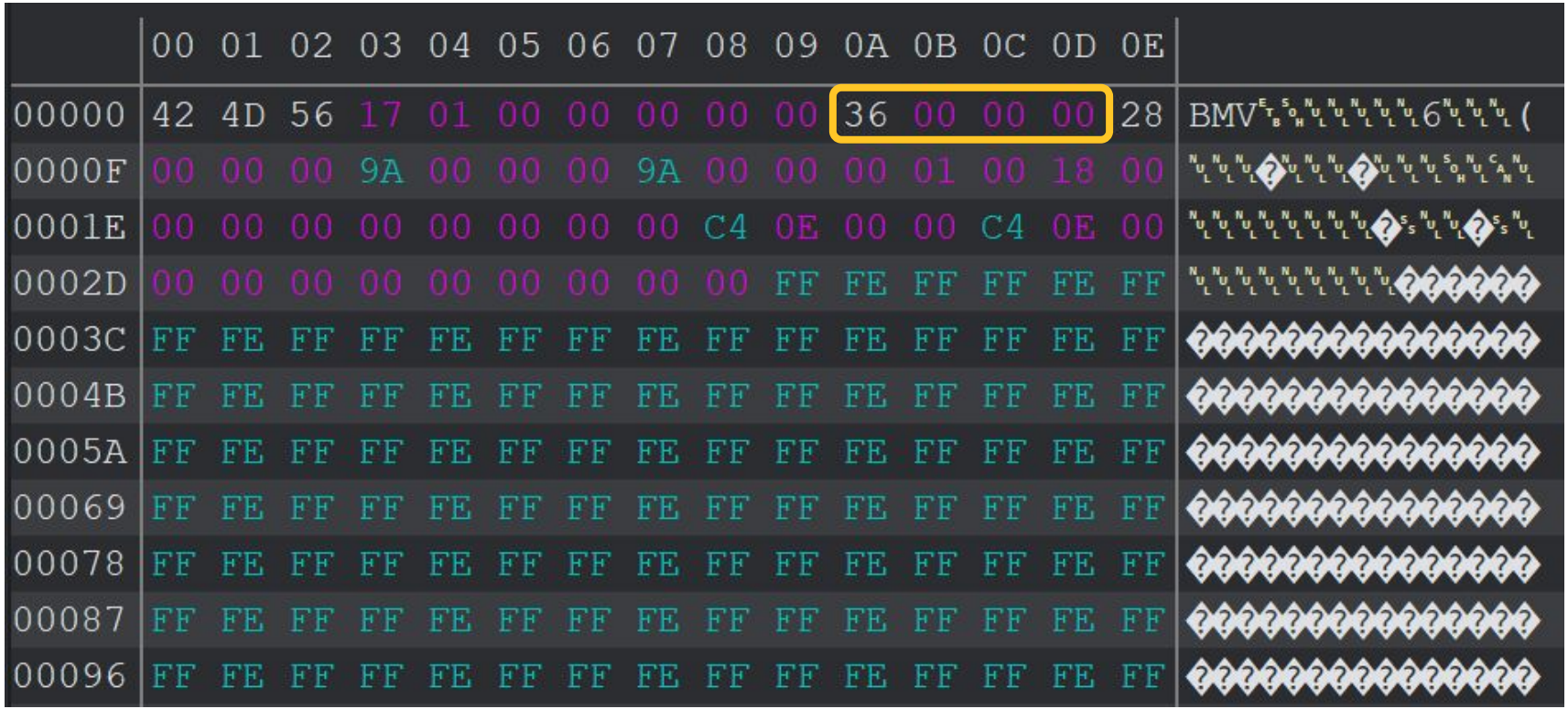


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

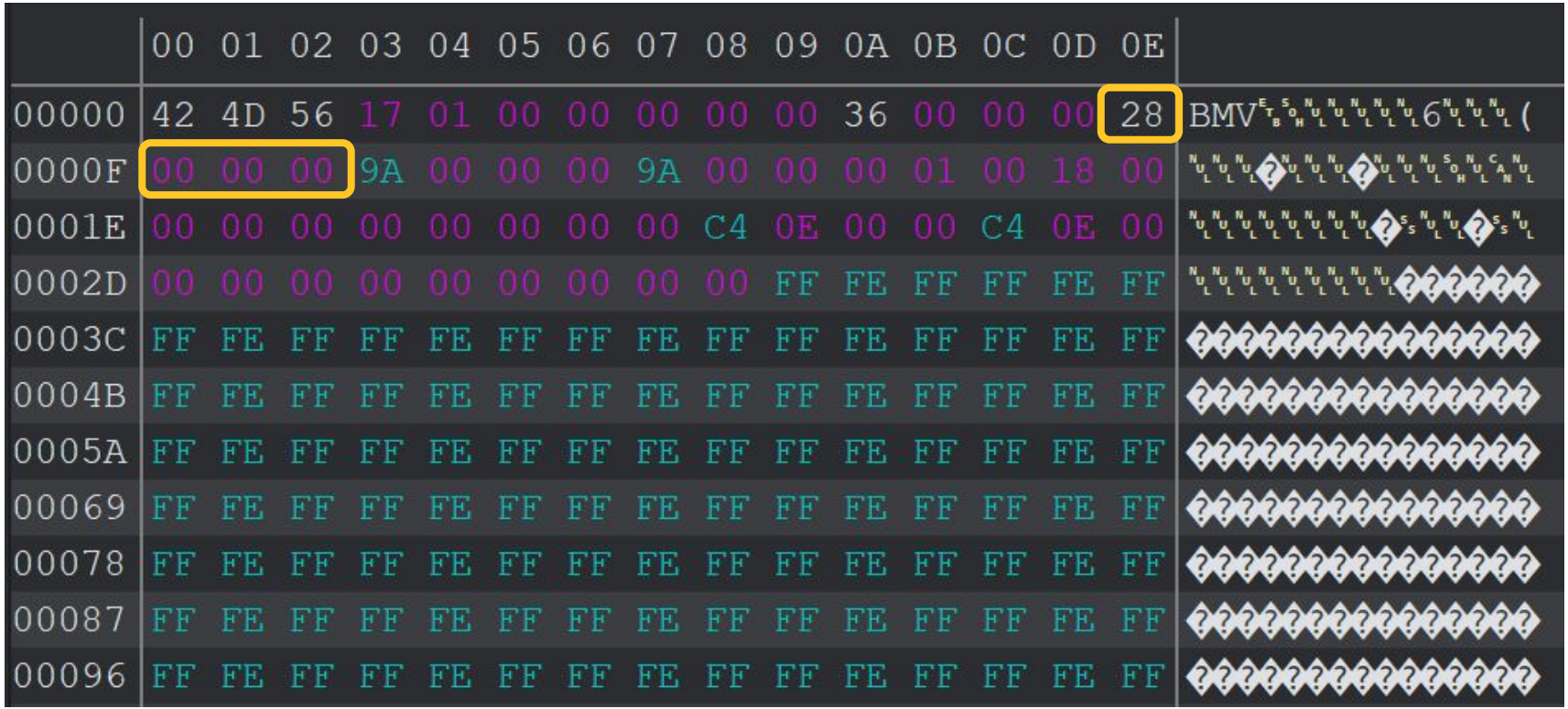


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

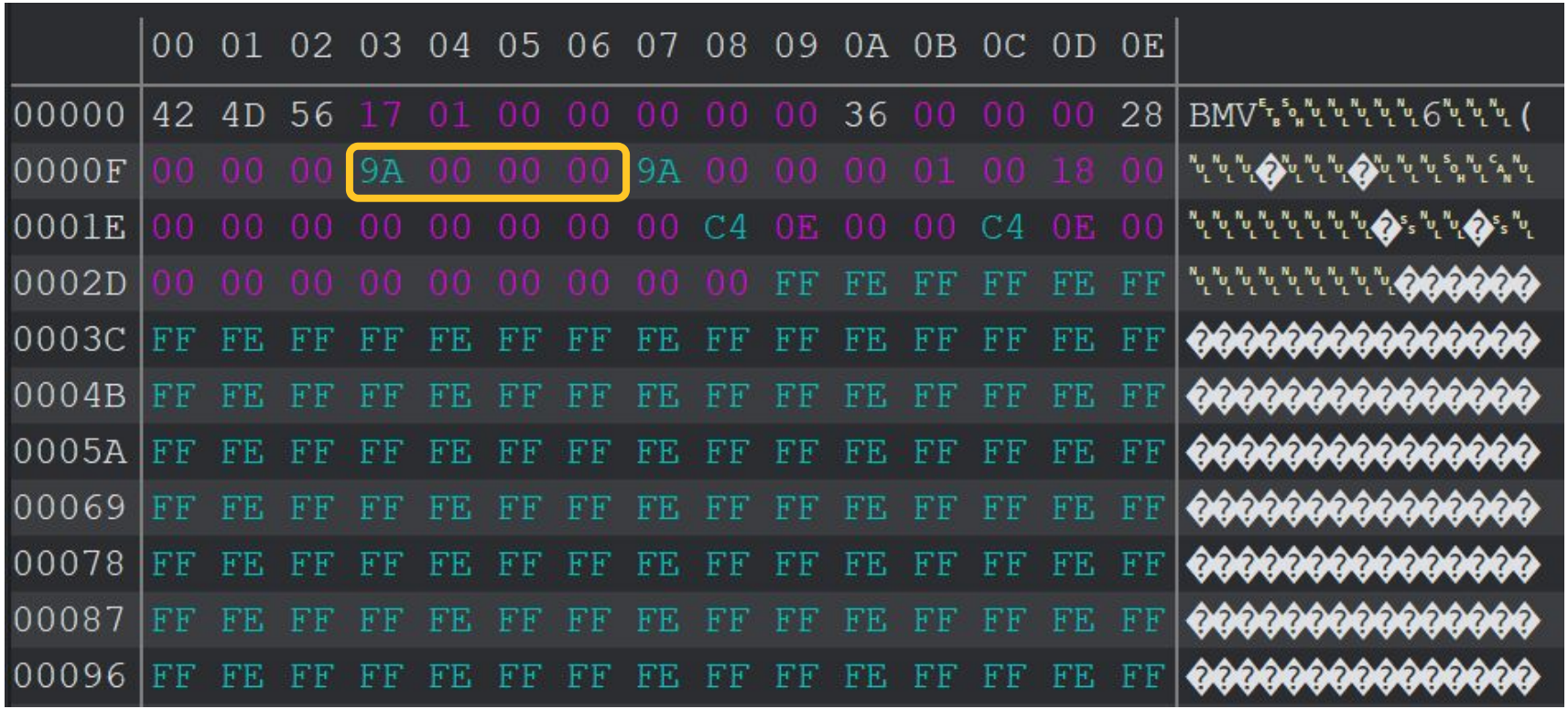


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

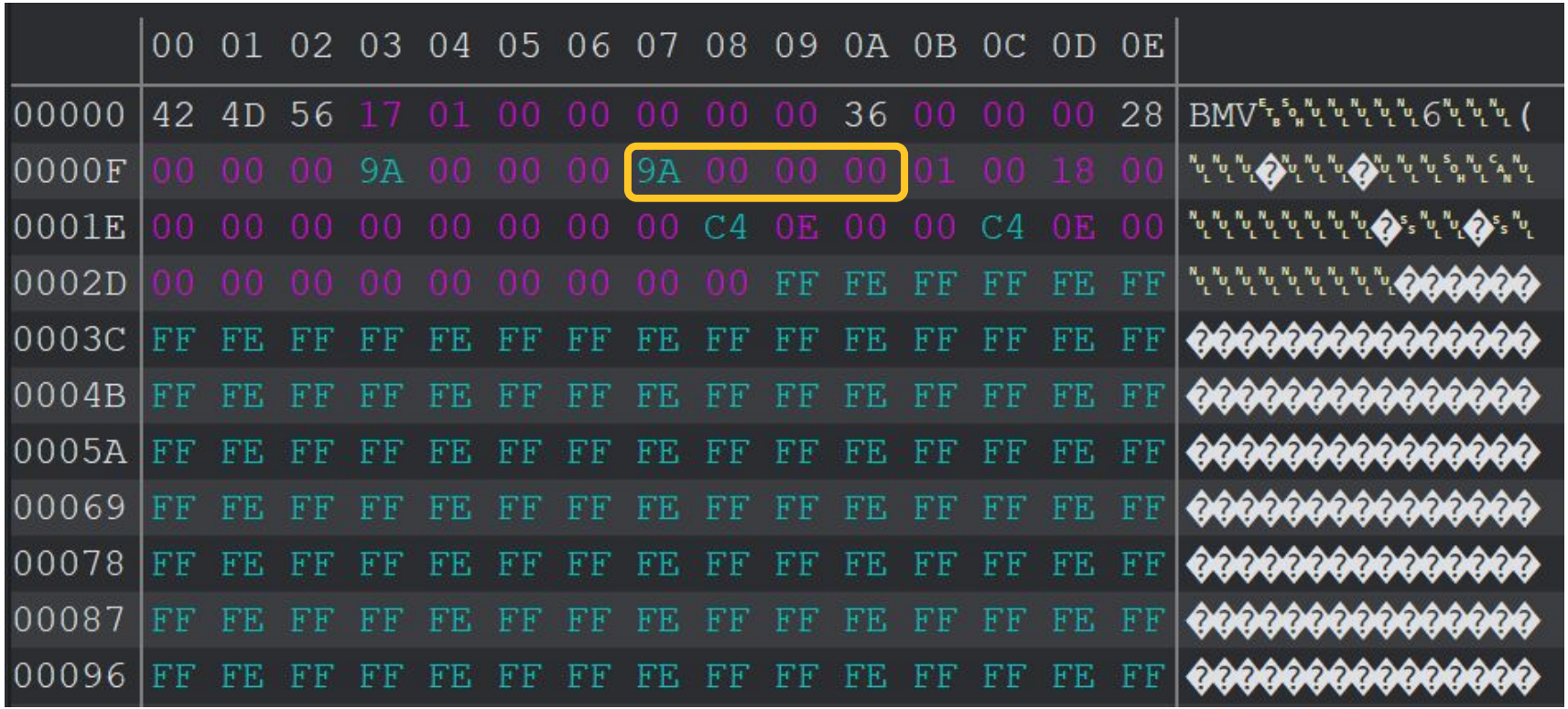


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

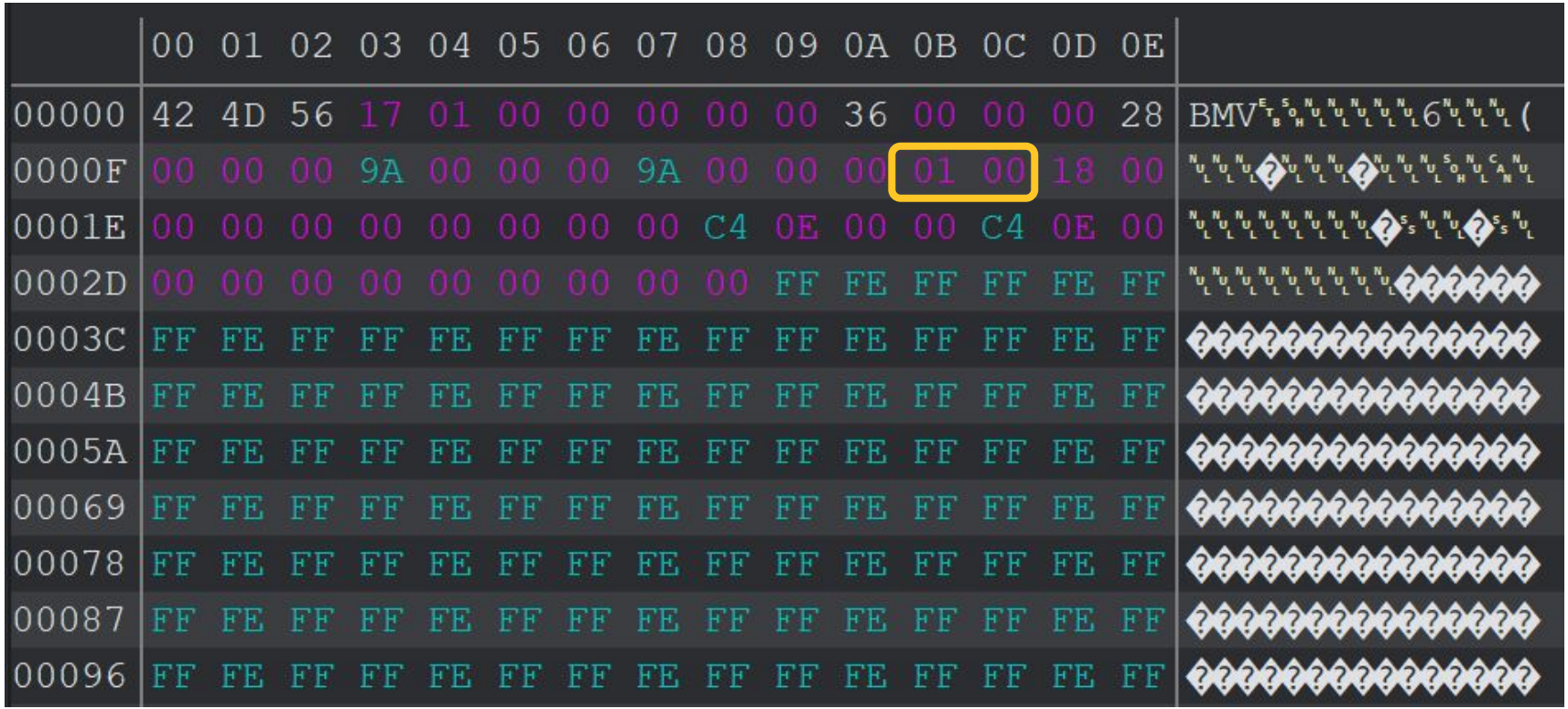


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

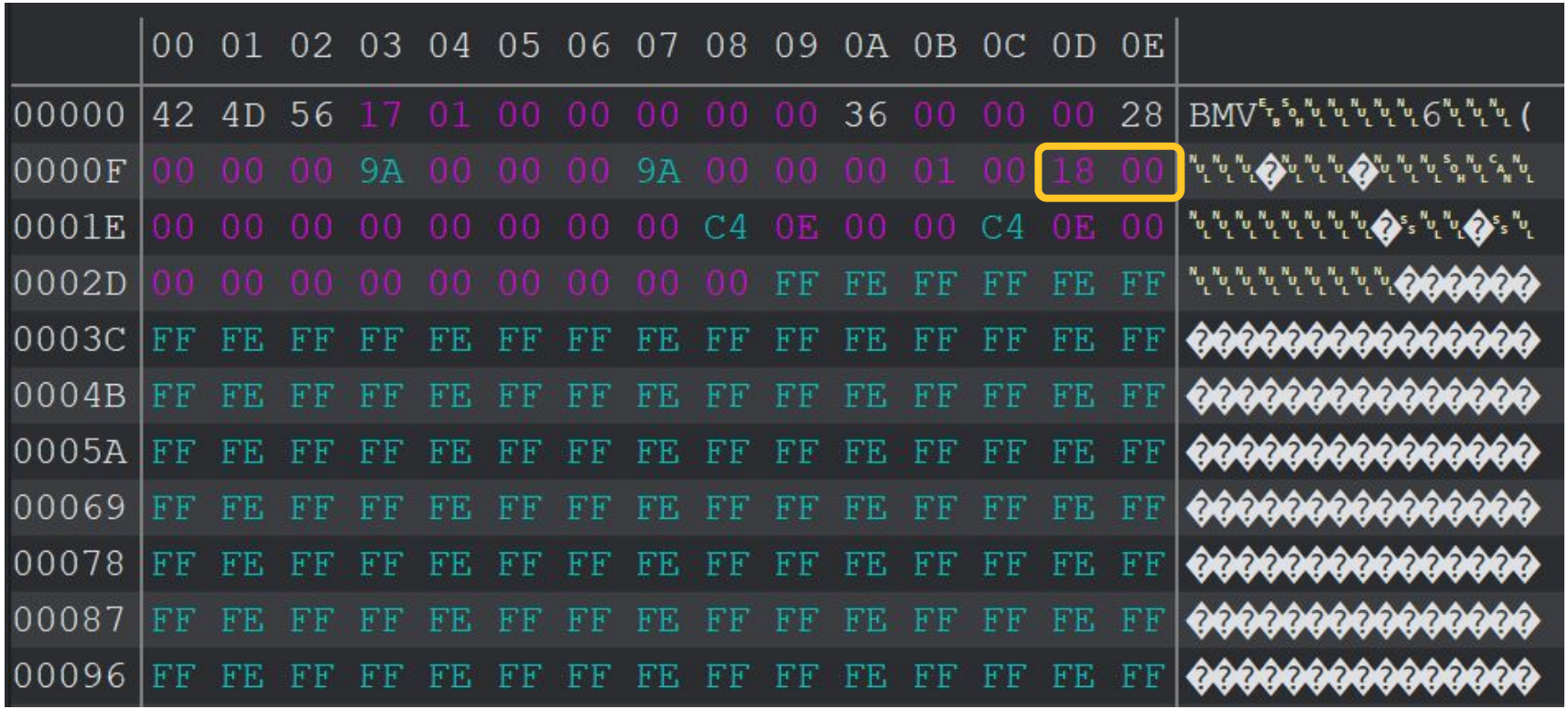


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

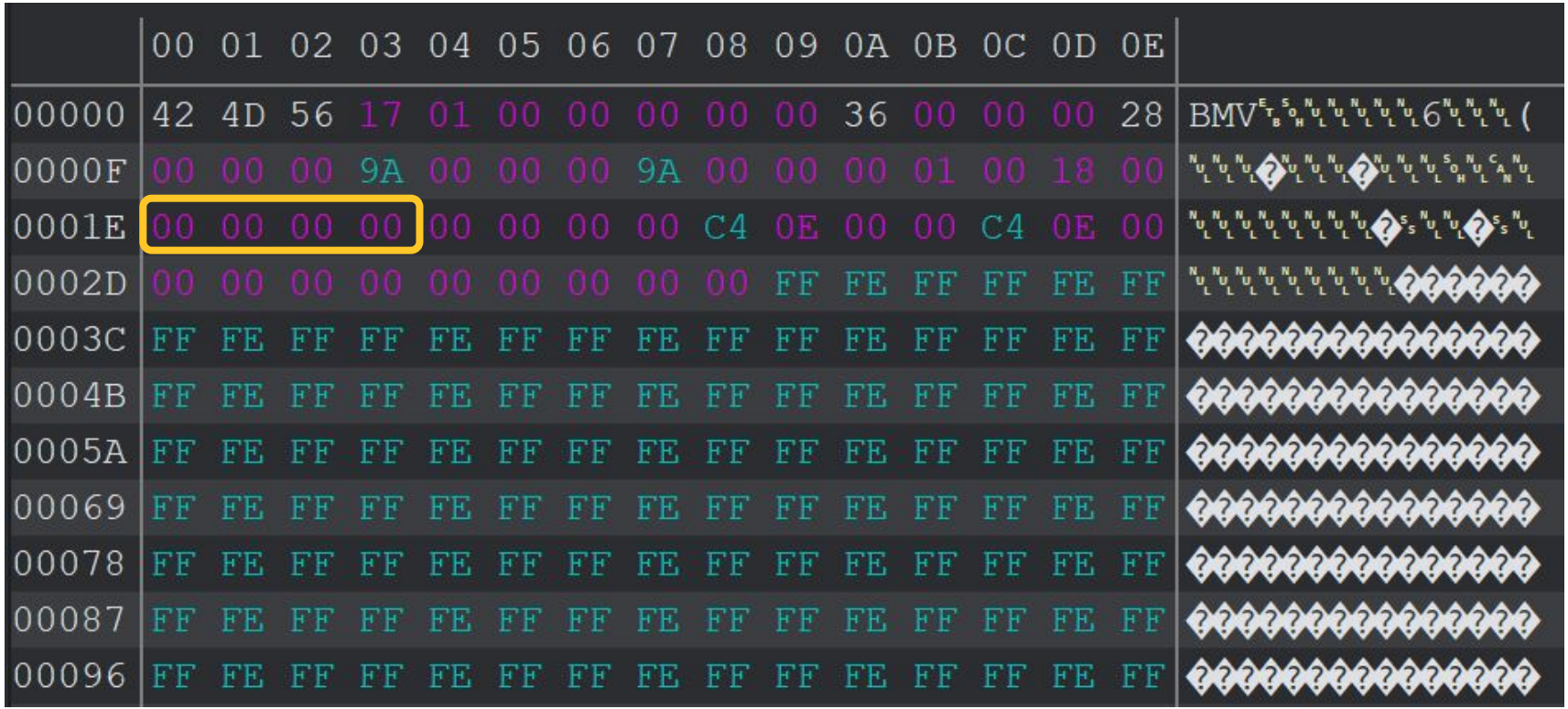


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

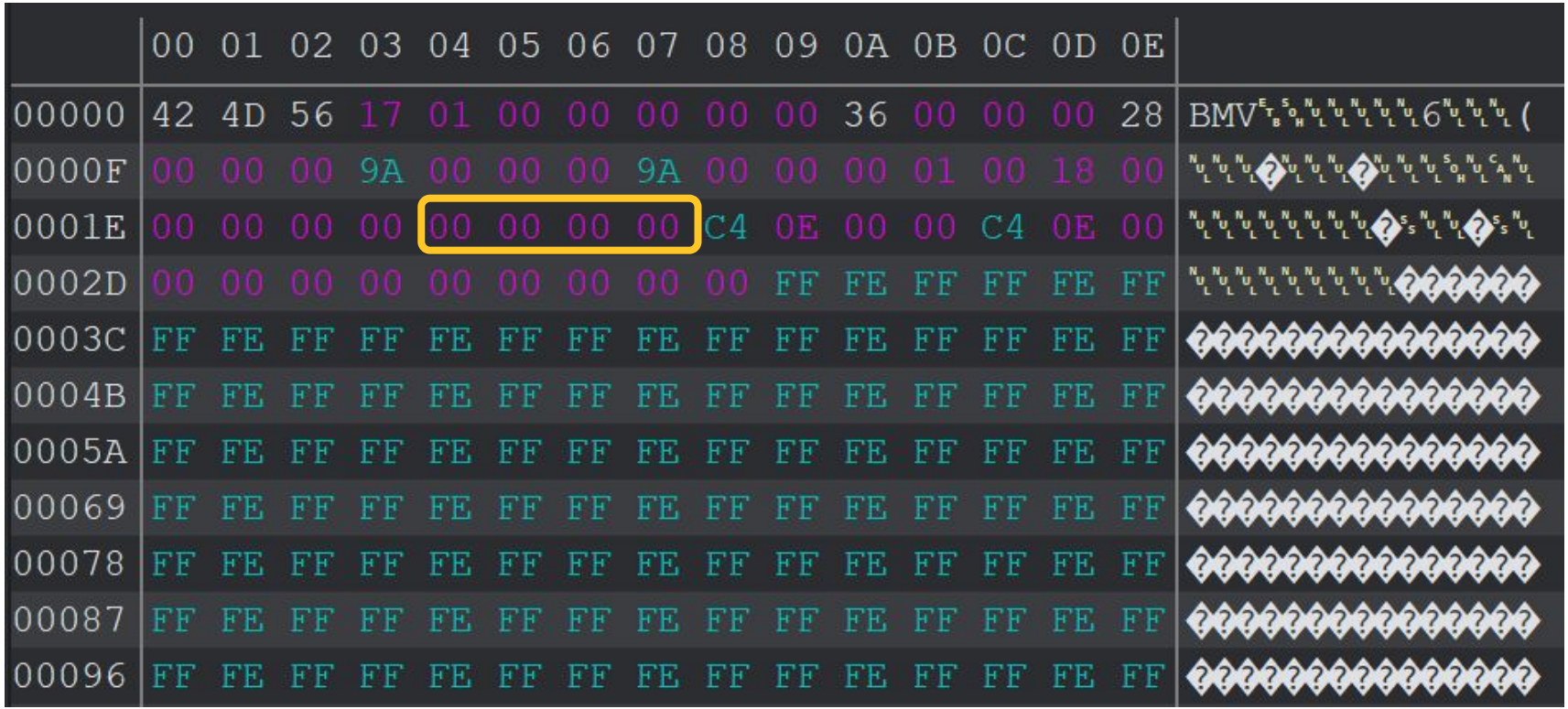


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

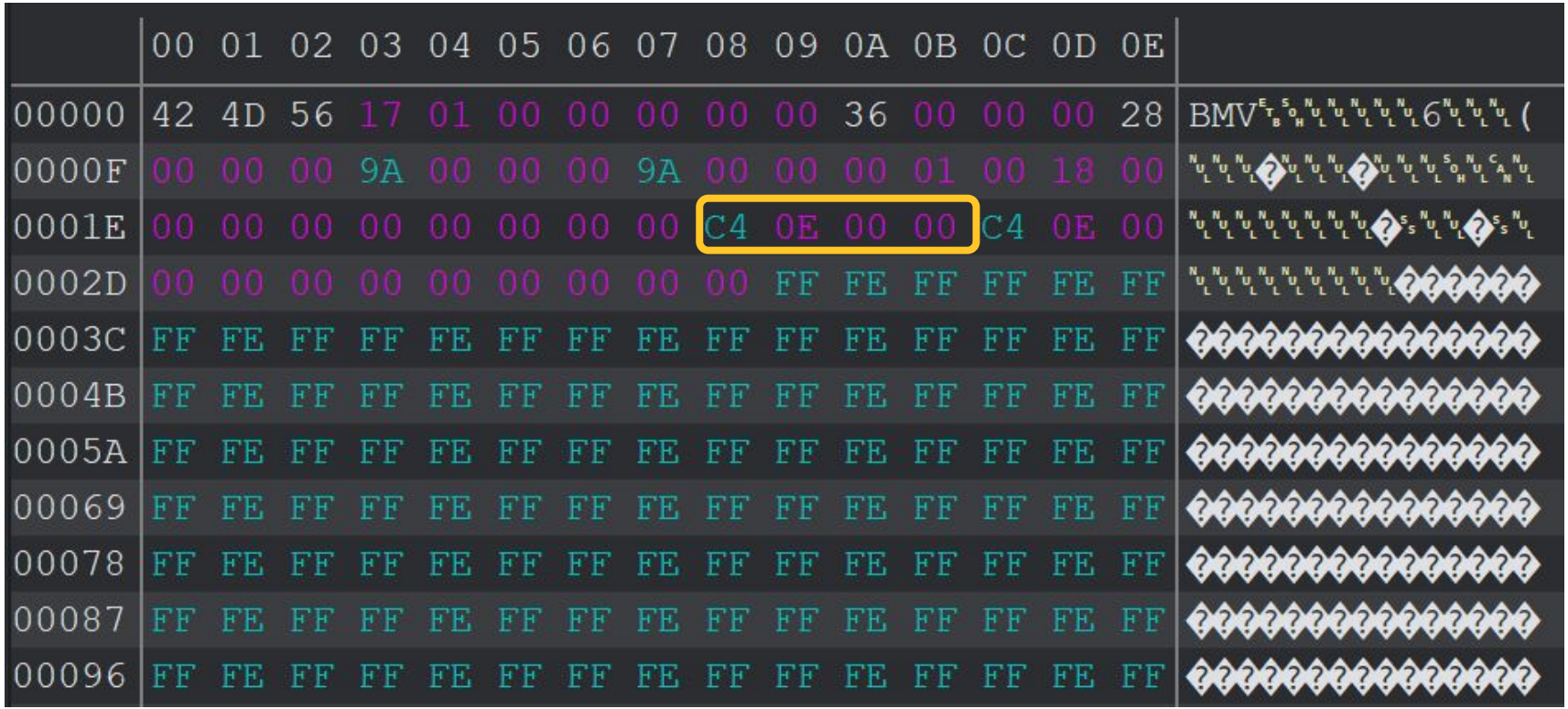


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

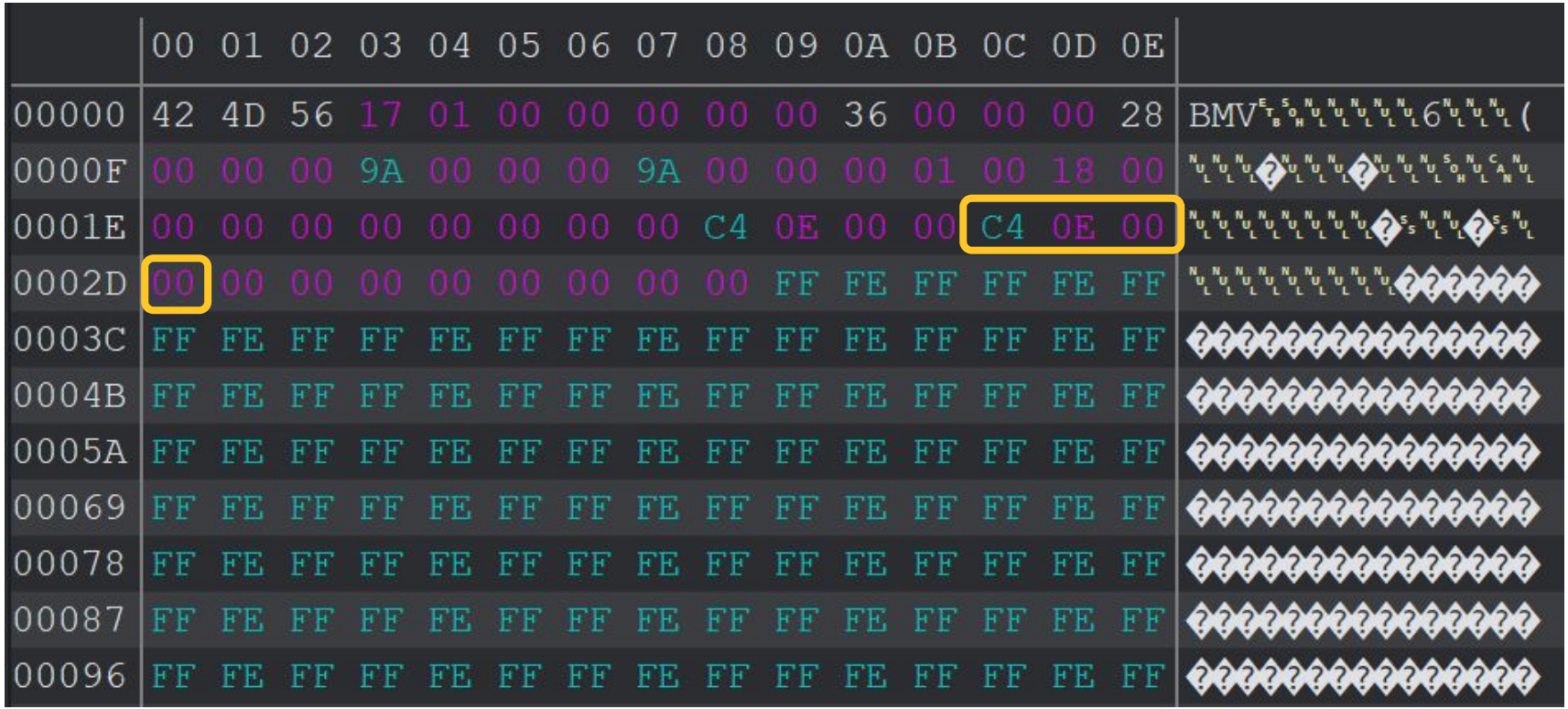


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

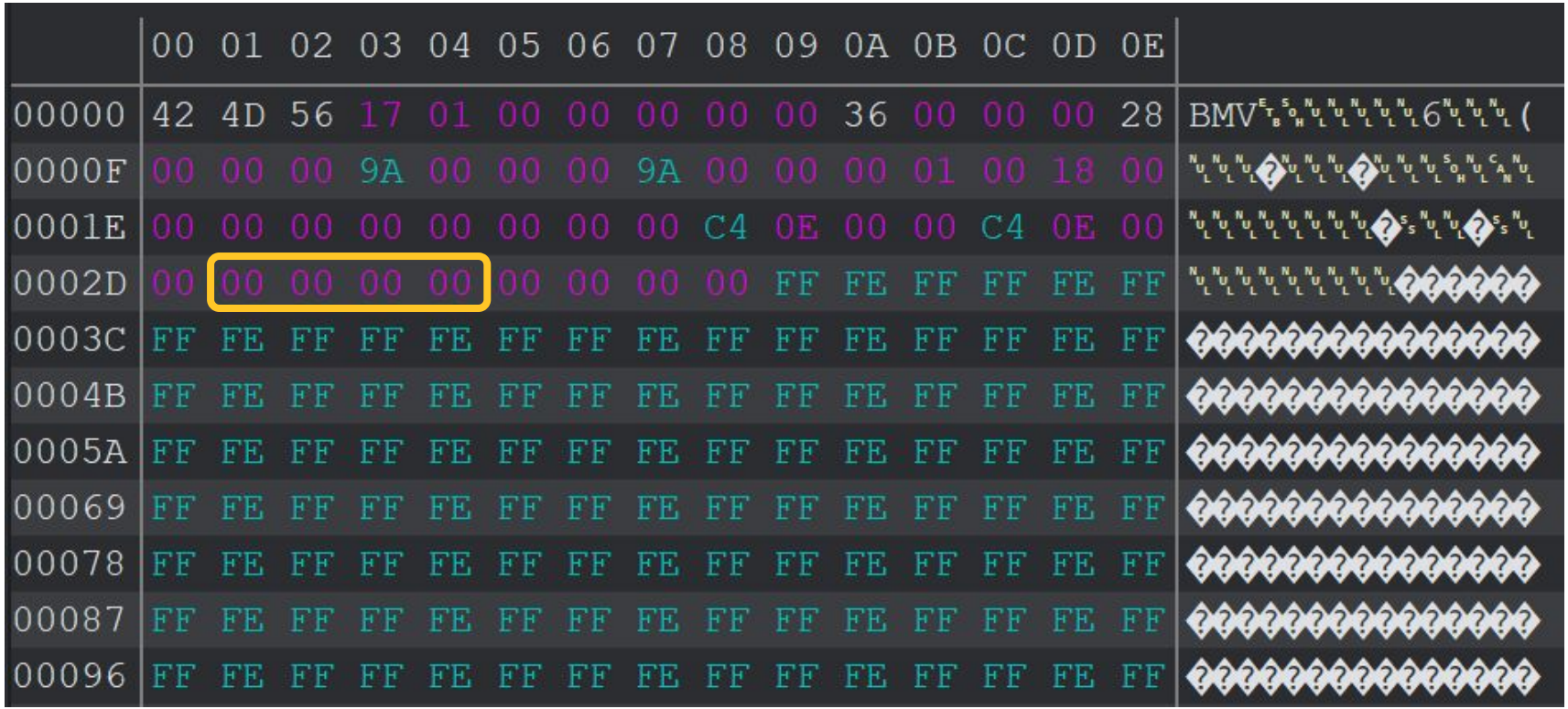


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?

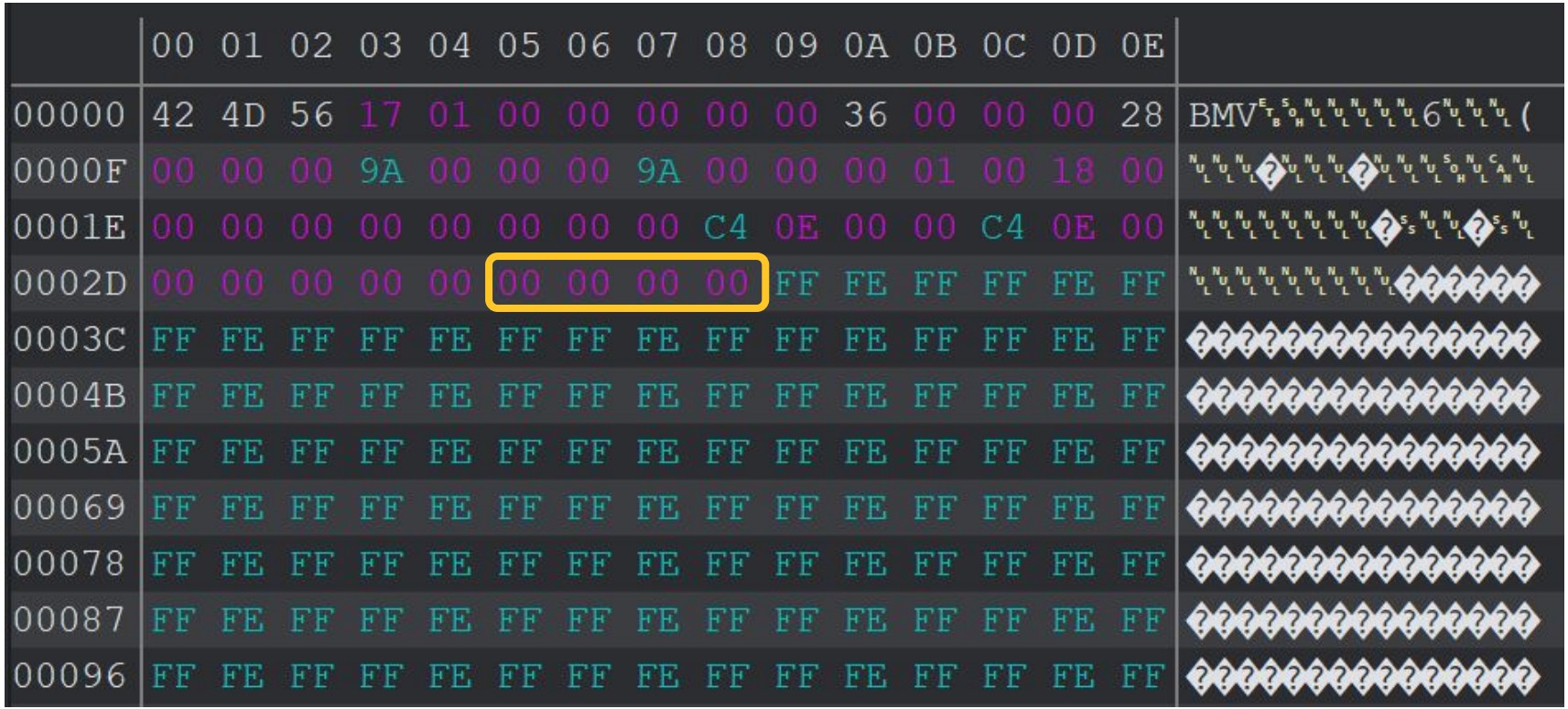


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

SER 334

BMP File Format

What is this piece?



	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000	42	4D	56	17	01	00	00	00	00	00	36	00	00	00	28	BMV 6 (
0000F	00	00	00	9A	00	00	00	9A	00	00	00	01	00	18	00	
0001E	00	00	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	
0002D	00	00	00	00	00	00	00	00	00	FF	FE	FF	FF	FE	FF	
0003C	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0004B	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
0005A	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00069	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00078	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00087	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	
00096	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	FF	FE	FF	

Upcoming Events

SI Sessions:

- Monday, January 22nd at 7:00 pm MST
- Sunday, January 28th at 7:00 pm MST **Cancelled - good luck on Exam 1!**
- Monday, January 29th at 7:00 pm MST

Review Sessions:

- Exam 1 Review: Thursday, January 25th 7:00 pm - 9:00 pm MST

Questions?

Survey:

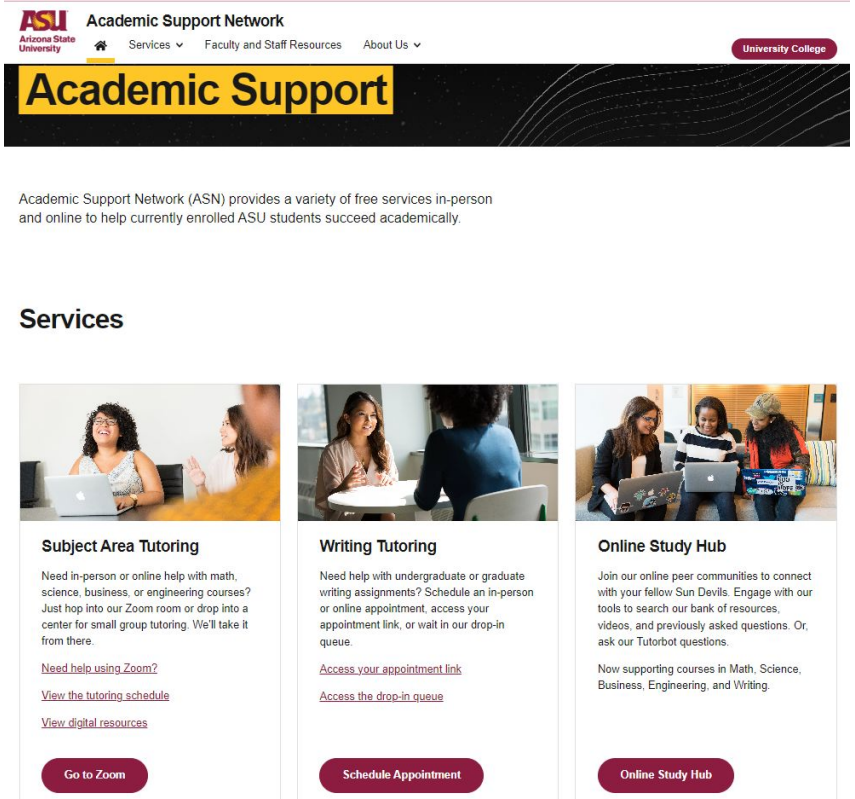
http://bit.ly/ASN2324

42

More Questions?

Check out our other resources!

tutoring.asu.edu

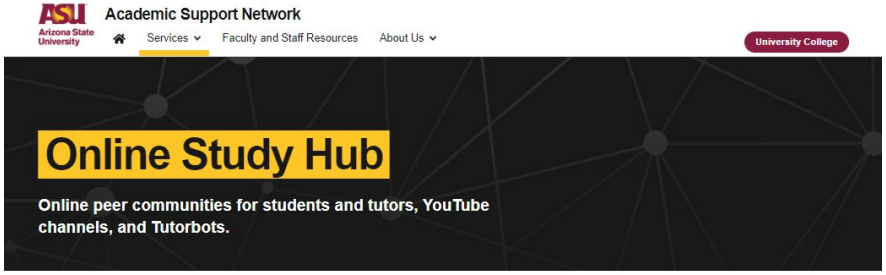


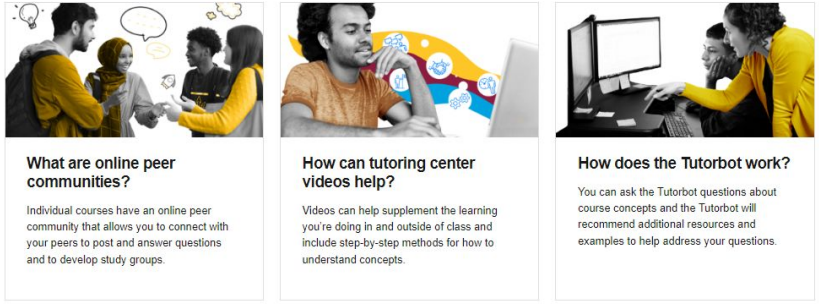
1. **Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.**
2. **Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.**

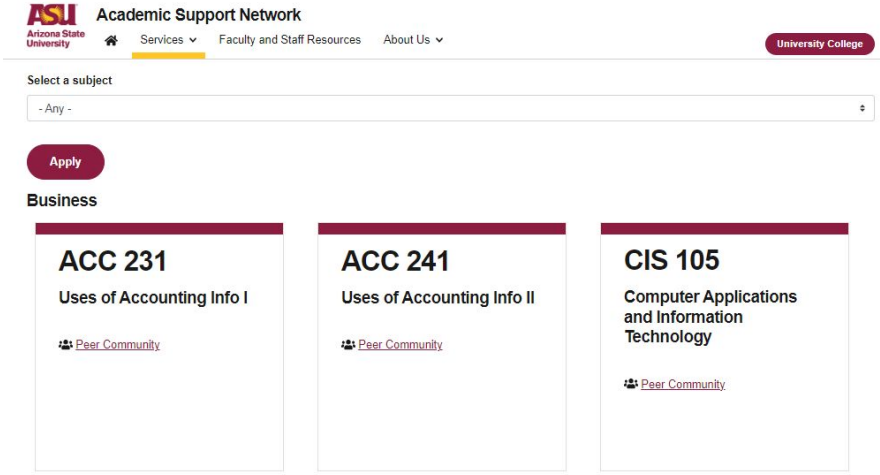
More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub







Don't forget to check out the Online Study Hub for additional resources!

Additional Resources

- Course Repo
- BMP File Format (Wiki)
- Linux Kernel API