Computer Science 230
Assignment 3 – Part I

Consider the following picutre of memory:

Memory Location	**Label**	**Memory Contents**
0x200	value1:	0xCD
0x201		0xAB
0x202	value2:	0x34
0x203		0x12
0x204	res:	
0x205		

That might have been generated by assembly directives (and some initialization code – that has been omitted) like:

```
.dseg
.org 0x200
value1:          .byte 2
value2:          .byte 2
res:             .byte 2
```

Given the AVR is little-endian, this memory picture reflects what might happen if you declared 16-bit variables like:

```
unsigned int value1 = 0xABCD;
unsigned int value2 = 0x1234;
unsigned int res;
```

Your job is to compute the sum: `res = value1 + value2`

You should use data-indirect addressing, assuming the following code has been executed to initialize the X, Y and Z registers to point to `value1`, `value2` and `res` respectively:

```
ldi XH,high(value1)
ldi XL,low(value1)
ldi YH,high(value2)
ldi YL,low(value2)
ldi ZH,high(res)
ldi ZL,low(res)
```

A reminder that you should add the least significant bytes first using the `ADD` instruction and then the most significant bytes using the `ADC` instruction.

Print page 2 of this document and bring it to class on Thursday February 26th. Alternatively, hand it in under my door (ECS 518) prior to 10:00 on February 26th.

Name : _____

Student Number : _____

Compute the sum:

```
      0xABCD
+     0x1234
------------
```

What is the least-signficant byte of the sum? _____

What is the most-significant byte of the sum? _____

Given the code on the previous page, write the sequence of instructions that will compute res = value1 + value2 using data indirect addressing: