# ARM Introduction

Email if you have any questions!

☐ I pledge my honor that I have abided by the Stevens Honor System

Turn audio ON

Time Remaining:

-1:57 Submit Quiz

## 1. Basic Knowledge (1 points)

Click to report a problem

How many integer registers are there on the Raspberry Pi given that it's in 32-bit mode? 16
Some of the registers are special and cannot be used as general registers.
For example register r15 is the PC, r14 is the SP, and r13 is LR.

## 2. Overflow (1 points)

Click to report a problem

```
Given that the maximum value for a 16 bit unsigned number is 65535
uint16_t x = 65533;
x += 3;
The result in x is 0.
A value is being computed:

int seconds = age * 365 * 24 * 60 * 60;

How might you detect that overflow is occurring?
☑answer is negative

☑answer grows smaller with increasing number

☑answer ends in a digit that is not zero

☐answer contains the digit 6

.
```

## 3. Basic Instructions (1 points)

Click to report a problem

```
After each of the following assembler instruction show the values in the registers
as 8 hex digits. (Do not put in 0x)
For the pc put in the last two digits
000102b4: mov  r0, #4    @r0=00000004    pc=000102b8
000102b8: sub  r0, #5    @r0=fffffff     pc=000102bc
000102bc: add  r0, #1    @r0=00000000    pc=000102c0
000102c0: add  r0, #1    @r0=00000001
```

## 4. Parameters (1 points)

Click to report a problem

```
Given the following function prototype, identify what register is used
for each parameter and return value
int f(int a, int b);

parameter a uses r0
```

parameter b uses r1
return value uses r0

5. calling functions (1 points) Click to report a problem

In ARM assembler, the bl (branch and link) instruction is used to call a function and bx (branch and exchange) instruction are used to return.

```
000104c8: mov  r0, #0    @r0=00000000    pc=000104cc
000104cc: add  r0, #4    @r0=00000004    pc=000104d0
000104d0: sub  r0, #3    @r0=00000001    pc=000104d4
000104d4: bl   000104f0  @lr=000104d8    pc=000104f0
000104d8: bl   000104e0  @lr=000104dc    pc=000104e0
000104dc: pop  {pc}  @*** your code ends here***
000104e0: mul  r0, #3    @r0=00000003    pc=000104e4
000104e4: bx   lr        @               pc=000104dc
000104e8: ldr  r1, =0xFFAB
000104ec: ldr  r2, =16384
000104f0: add  r0, #1    @r0=00000001
000104f4: bx   lr        @               pc=000104d8
```

Time Remaining: -1:57 Submit Quiz