# Practice Robotics Exam

NOTE: The exam next week is scheduled for 50 minutes. The time limit is a part of the challenge for scaling purposes. The end result will be scaled. So, if the average mark in the class is say 35% that is a C result.

NOTE 2: The exam is scheduled in the second session of our double. If you are in line 2 Cyber Security we can either attempt to do it in line 2 or you can schedule to do it in any of my teaching lines between Monday and Wednesday. I am happy to get you excused from a different class.

Scores calculation:

1 point questions:

| 0 points | 1 point |
| --- | --- |
| Answer was not submitted or was not able to be assessed | Answer is given and meet expectations for knowledge or understanding on the topic |

2 point questions:

| 0 | 1 | 2 |
| --- | --- | --- |
| Answer was not submitted or was not able to be assessed | Answer is given but does not meet expectations for knowledge or understanding on the topic | Answer is given and meet expectations for knowledge or understanding on the topic |

4 point questions:

| 0 | 1 | 2 | 3 | 4 |
| --- | --- | --- | --- | --- |
| Answer was not submitted or was not able to be assessed | Answer was incomplete and showed a limited understanding of the topic space (1 or 2 of 5 (simple) bugs found) | The answer submitted was a partial response however it showed a growing understanding of the topic space (>2 bugs found ) | The answer submitted was a partial response but answered the majority of the question (>=4 bugs found) | Answer submitted is complete (>=5 bugs found) |

## Scope of questions:

* Variables and conditions
* Functions
* Iteration arrays and (a tiny amount of) pointers
* Arduino signals and systems

### MC: What is the result of the following code? (1 mark)

int a = (8-3)/(5\*\*2/4)  
printf("%f\n",a);

1. 0.8
2. 7.97
3. “5/2.5”
4. 2

### MC: Consider the following code block and select the most accurate answer (1 mark)

#include <stdio.h>  
  
void hit(char \* attacker, char \* defender, char \* weapon, int damage){  
 printf("%s hit %s with a %s for %d points of damage", attacker, defender, weapon, damage);  
}  
  
int main(void) {  
 hit("ada", "bob", "sword", 8);  
 hit("charles", "erin", "mace", 8);  
 return 0;  
}

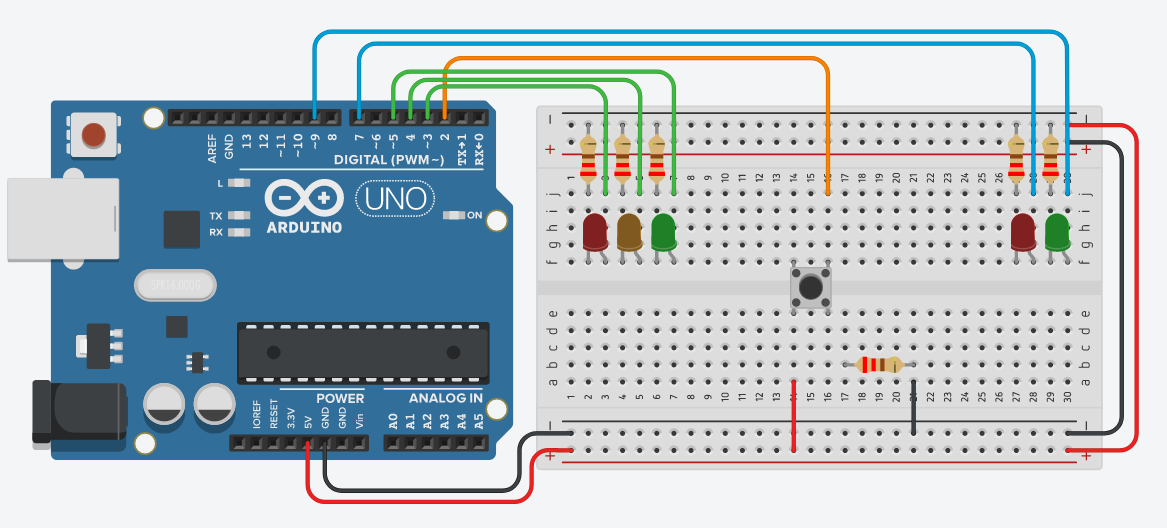
1. ada hit bob with a sword for 8 points of damage  
   charles hit erin with a mace for 8 points of damage
2. ada hit bob with a sword for 8 points of damage\n  
   charles hit erin with a mace for 8 points of damage\n
3. This code will not work as intended because lines 4 and 5 have integers and they should be strings
4. This code will not compile because the hit function doesn’t return a string / character array

### Short Answer: What is the likely output of the following code and why? (2 marks)

int main(void) {  
 int j = 1;  
 int c = j;  
 while (c > 0) {  
 for (int i = 0; i < c; i++){  
 printf("\*");  
 }  
 printf("\n");  
 if (c == 5) j = 0 - 1;  
 c += j;  
 }  
 return 0;  
}

1. This is clearly the work of witchcraft
2. This will likely have an exception due to the if (c == 5) j = 0 - 1; being nonsense
3. This code attempts to create a triangle that is 9 high and 5 across but it is broken by the while condition never ending
4. This code attempts to create a triangle that is 9 high and 5 across and is successful

### Long answer: There are at least 5 errors in the following circuit and code. (4 marks)



#define TRAFFIC\_RED trafficLights[2]  
#define TRAFFIC\_YELLOW trafficLights[1]  
#define TRAFFIC\_GREEN trafficLights[0]  
  
#define PED\_RED pedLights[0]  
#define PED\_GREEN pedLights[1]  
  
#define TIME\_UNIT 1000  
  
int button =2;   
  
int trafficLights[] = {5, 4, 3};  
int pedLights[] = {8, 9};  
  
int buttonState = 0;  
void setup()  
{  
 Serial.start(9600);  
 Serial.println("Initialising Traffic Lights...");  
 pinMode(button, INPUT);  
   
 for (int i = 0; i < sizeof(trafficLights); i++)  
 {  
 pinMode(trafficLights[i], OUTPUT);  
 }  
 for (int i = 0; i < sizeof(pedLights); i++)  
 {  
 pinMode(pedLights[j], OUTPUT);  
 }  
   
 digitalWrite(TRAFFIC\_GREEN, HIGH);  
 digitalWrite(PED\_RED, HIGH);  
   
 Serial.println("Initialisation complete");  
}  
  
void phaseLights(bool lightOff, bool lightOn){  
 digitalWrite(lightOff, LOW);  
 digitalWrite(lightOn, HIGH);  
}  
  
void flashLight(int light, int d){  
 digitalWrite(light, LOW);  
 delay(d);  
 digitalWrite(light, HIGH);  
 delay(d);   
}  
  
void trafficAllGreen(){  
 digitalWrite(TRAFFIC\_GREEN, HIGH);  
 digitalWrite(PED\_RED, HIGH);  
}  
  
void trafficAllRed(){  
 delay(TIME\_UNIT);  
 phaseLights(TRAFFIC\_GREEN, TRAFFIC\_YELLOW);  
 delay(TIME\_UNIT\*3);  
 phaseLights(TRAFFIC\_YELLOW, TRAFFIC\_RED);  
 delay(TIME\_UNIT\*5);  
}  
void pedAllGreen(){  
 phaseLights(PED\_RED, PED\_GREEN);  
 delay(TIME\_UNIT\*5);  
 phaseLights(PED\_GREEN, PED\_RED);  
 for (int i = 0; i < 10; i++)  
 {  
 flashLight(PED\_RED, TIME\_UNIT);  
 }  
   
}  
  
void loop()  
{  
 buttonState = digitalRead(button);  
 if (buttonState);  
 {  
 trafficAllRed();  
 pedAllGreen();  
 trafficAllGreen();  
 }  
   
}

answer1:   
  
  
answer2:   
  
  
answer3:  
  
  
answer4:  
  
  
  
answer5: