Performance Task: Create - Applications from Ideas Written Responses

Entire Project

Computational Artifact

AP Computer Science Principles Explore Performance Task

Rafe-Gerson-AP-Computer-Science-Principles-Performance-Tasks-Repository

Program Purpose and Development

2a. *Provide a written response or audio narration in your video that: identifies the programming language, identifies the purpose of your program, and explains what the video illustrates.* My "Ultimate RGB Arduino Mining Rig aka Fire-Alarm Prank-Bot" was developed in Arduino, a native c/c++ IDE for various robotics boards, and is intended to humor me and my peers. Very much like the "downloading RAM" fiasco, my program does not generate anything and its real functions are: controlling the color scheme of an RGB bulb via for-loops, printing pseudo mining hash-rates to an LCD display, recording the temperature of the Arduino, entertaining and/or pestering nearby individuals with a "melody" of my own making, which sounds awfully like that of a fire-alarm, and communicating other data to the host computer's Serial Monitor. You can view all of these functions play out in the video.

2b. *Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.* The main difficulty I encountered in the incremental and iterative development process of my program was implementing all of the components I had planned. The Breadboard itself only had so many available "pores," and the main Arduino Board only possessed two electrical outputs--a 5 volt and a much weaker 3.3 volt slot--and only a few GND slots which meant that I could only power two aspects of the designs. Luckily, I scrounged a 9 volt battery from my father's "pile of stuff" which enabled me to generate more than enough power for all of the pieces via a Power Supply Module.

In order to successfully implement all of the selected components, I had to read up on all of the Arduino functions and libraries I would necessitate in order to attain the desired results. Functions directed at the LCD display began with "lcd.", functions directed at the RGB bulb commenced with "analogWrite()", data communicated to the host computer began with "Serial.", and all of the components had to be alloted pin slots via the "#define" command. The pre-implemented libraries--indicated by the "include" function--and various c/c++ headers really facilitated my independent program development.

2c. *Capture and paste a program code segment that implements an algorithm and that is fundamental for your program to achieve its intended purpose. This code segment must be an*

algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program.

```
for (int i = 0; i < 255; i += 1) {
   redValue -= 1;
   greenValue += 1;
   analogWrite(RED, redValue);
   analogWrite(GREEN, greenValue);
 redValue = 0;
 greenValue = 255;
 blueValue = 0;

 for (int i = 0; i < 255; i += 1) {
  greenValue -= 1;
  blueValue += 1;
  analogWrite(GREEN, greenValue);
  analogWrite(BLUE, blueValue);
  delay(delayTime);
 }

 redValue = 0;
 greenValue = 0;
 blueValue = 255;

 for (int i = 0; i < 255; i += 1) {
  blueValue -= 1;
  redValue += 1;
  analogWrite(BLUE, blueValue);
  analogWrite(RED, redValue);
  delay(delayTime);
 }
```

The functions depicted above cycle through all of the values on the RGB spectrum. A single variant of one of the above for-loops would increase or decrease one RGB channel, which would yield only one color on the RGB bulb by default. When all of the for-loops function in conjunction, the RGB bulb appears to cycle through all of the colors of the rainbow, very much like your standard RGB PC.

2d. *Capture and paste a program code segment that contains an abstraction you developed individually on your own. This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program.*

```
int PHrate = (random(100, 999));
```

The latter code manages the complexity of the program as it renders the "hash-rates" similar. Instead of utilizing the "random" command for both the Serial Monitor and the LCD display and possibly receiving different numbers, setting them both to a variable would allow me to generate an arbitrary number that is consistent for the two.

3. **Program Code**

||
V

Let the blue ovals represent the segment of program code that implements the algorithm I created for my program that integrates other algorithms and integrates mathematical and/or logical concepts.
Let the yellow rectangles represent the segment of program code that represents an abstraction I developed.

```
#include <Wire.h> //built-in to the Arduino IDE
#include <LiquidCrystal.h> //built-in to the Arduino IDE
#include "pitches.h" //acquired from https://gist.github.com/mikeputnam/2820675
#define BLUE 2 //PWM slot 2 - blue channel
#define GREEN 3 //PWM slot 3 - green channel
#define RED 4 //PWM slot 4 - red channel
//5 is occupied by passive buzzer
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //LCD pins

int i; //for control purposes

int tempPin = 0;

int melody[] = {
  NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7
};

int duration = 500; //constant integer for the "music"

void setup() {
  Serial.begin(9600);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
```

```
  lcd.begin(16, 2); //max number for lcd displays
  lcd.print("Setting up Miner");
  delay(500);
  Serial.println("Initialize DS3231");
  //The below print statements are just for fun! Have fun translating them!
  Serial.println("Настройка шахтёра.");
  delay(500);
  Serial.println("Скачивание большего количества оперативной памяти из полностью открытого
исходного кода, 100-процентного бесплатного веб-сайта memes 'downloadmoreram.com'!");
  delay(500);
  Serial.println("RAM успешно запущена, я имею в виду скачанный.");
  delay(500);
  Serial.println("У вас есть оперативная память!");
  delay(500);
  Serial.println("Congratulations!");
  delay(500);
  Serial.println("Configuring miner.");
  delay(500);
  Serial.println("Downloading more VRAM from 'downloadmorevram.com'!");
  delay(500);
  Serial.println("Successful.");
  delay(500);
  Serial.println("You have downloaded 32 quadrillion petabytes of VRAM!");
  Serial.println("Look at that, you can finally get 60 frames per second on Fortnite!");
  delay(500);
  Serial.println("Доступ к Биткойнному кошельку!");
  delay(500);
  Serial.println("Перевод в СССР!");
  delay(500);
  Serial.println("Все ваши биткойны ушли!");
  delay(500);
  Serial.println("Но вы не знаете, что, если вы не перевели эту программу!");
  delay(500);
  Serial.println("Accessing personal information.");
  delay(100);
  Serial.println("Accessing cryptocurrency wallet.");
  delay(500);
  Serial.println("Transfer initiated.");
  delay(500);
}

int redValue;
int greenValue;
int blueValue;

void loop() {
  int PHrate = (random(100, 999));
```

```
#define delayTime 5
  redValue = 255;
  greenValue = 0;
  blueValue = 0;
  lcd.setCursor(0, 1);
  lcd.print("Elapsed: ");
  lcd.println(millis() / 1000);
  lcd.setCursor(0, 0);
  lcd.print(PHrate);
  lcd.print(" PH/s        ");

  int tempReading = analogRead(tempPin);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
  tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK )) * tempK );
  float tempC = tempK - 273.15;
  float tempF = (tempC * 9.0) / 5.0 + 32.0; //acquired from https://ideone.com/fork/tS9IX1

  Serial.print("Mining rig temperature in Celsius is ");
  Serial.println(tempC);
  Serial.print("Mining rig temperature in Fahrenheit is ");
  Serial.println(tempF);

  Serial.print(PHrate);
  Serial.println(" PH/s");
  Serial.print("Time elapsed in seconds is ");
  Serial.println(millis() / 1000);
  Serial.println();

  for (int thisNote = 0; thisNote < 8; thisNote++) {
    tone(5, melody[thisNote], duration);
    delay(1000);
  }

  for (int i = 0; i < 255; i += 1) {
    redValue -= 1;
    greenValue += 1;
    analogWrite(RED, redValue);
    analogWrite(GREEN, greenValue);
    delay(delayTime);
  }
```

```
  redValue = 0;
  greenValue = 255;
  blueValue = 0;

  for (int i = 0; i < 255; i += 1) {
    greenValue -= 1;
    blueValue += 1;
    analogWrite(GREEN, greenValue);
    analogWrite(BLUE, blueValue);
    delay(delayTime);
  }

  redValue = 0;
  greenValue = 0;
  blueValue = 255;

  for (int i = 0; i < 255; i += 1) {
    blueValue -= 1;
    redValue += 1;
    analogWrite(BLUE, blueValue);
    analogWrite(RED, redValue);
    delay(delayTime);
  }
}
```