+Requires the following:

1. Micro SD card
2. Any necessary adapters to connect micro SD to computer
3. Power supply through usb-c for raspberry pi 4 (micro b for previous models) 5V @ 1A (2A+ is preferred)
4. Computer
5. Raspberry Pi Imager download here: https://www.raspberrypi.org/downloads/
6. Ethernet cable and any necessary adapters
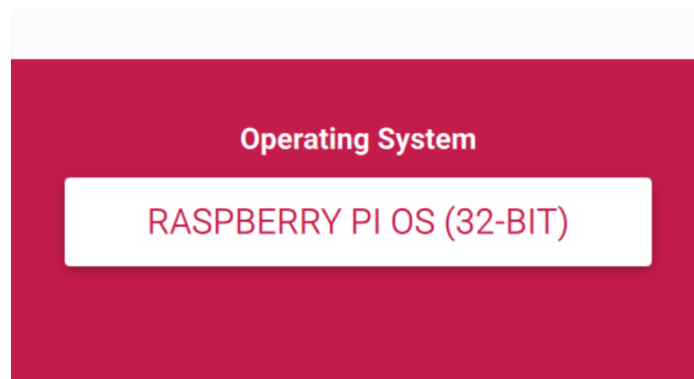7. Raspberry pi
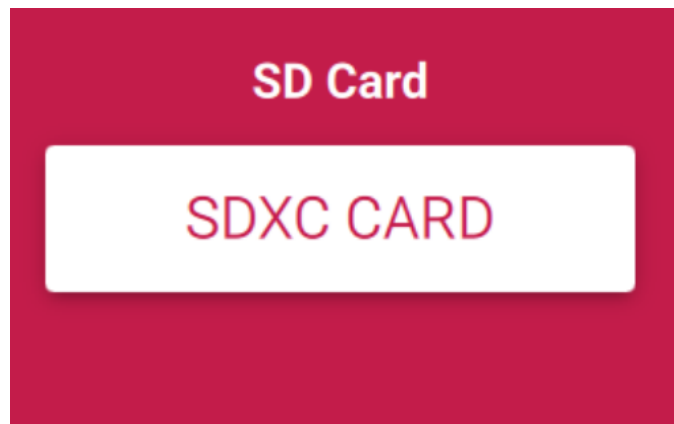
Step 1: Download the Raspberry Pi Imager

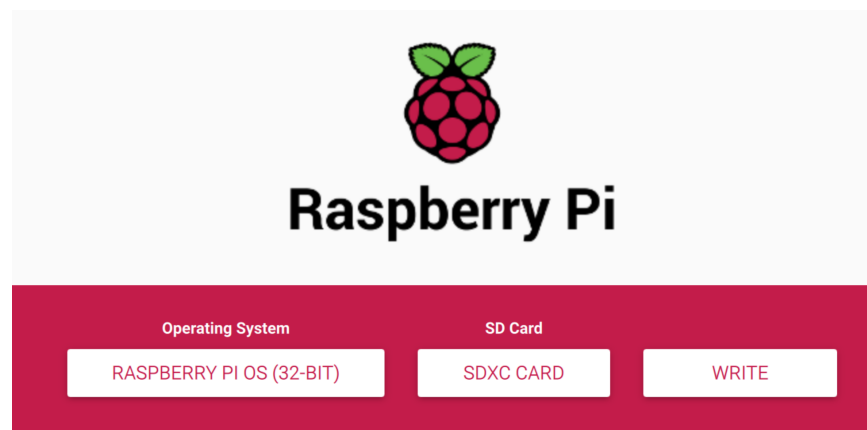Step 2: Insert SD card with Micro SD into Computer:
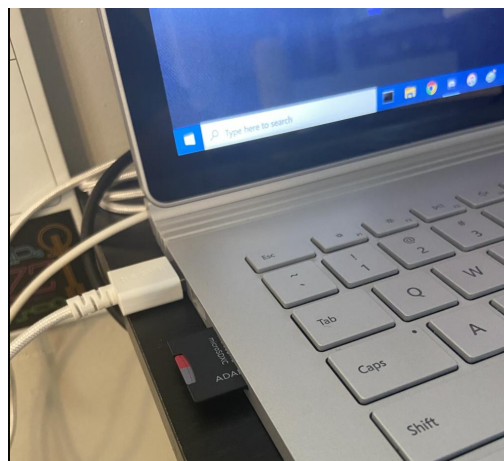
Step 3: Choose Operating System - RASPBERRY PI OS (32-BIT)

Step 4: Choose Storage device (Micro SD card)

Step 5: Click on "WRITE"
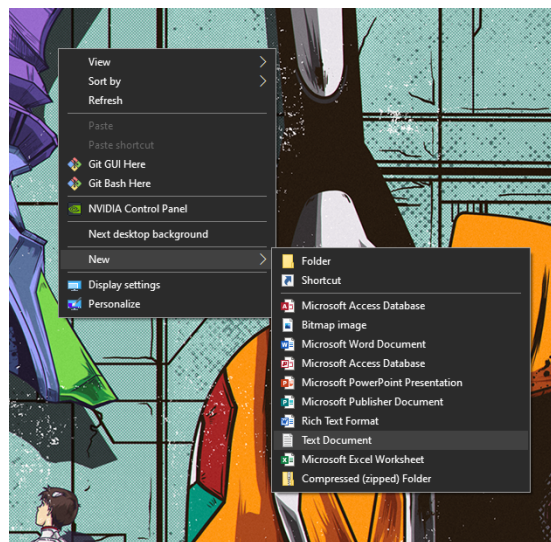
You should see the following:

Please be patient with write time

OS setup is comp[lete but keep the SD card connected to your computer

Step 6: Enable SSH (secure shell).

1. Create a .txt file (text file) by right clicking on the desktop:

2. Rename the text file SSH and save it as ALL files type, not .txt type

3. Open up the micro SD Storage file in file explorer:

4. Place SSH file into boot folder inside of micro SD files
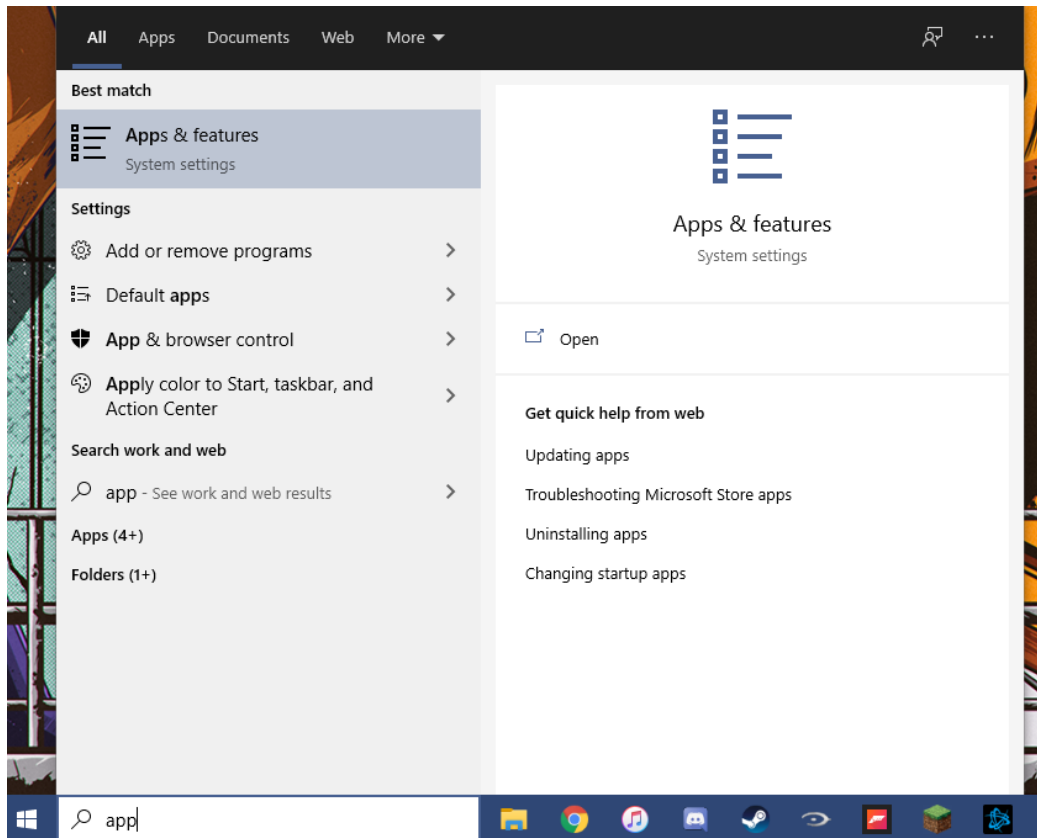
Step 8: Eject Micro SD storage from computer and plug it into raspberry pi (unpowered)
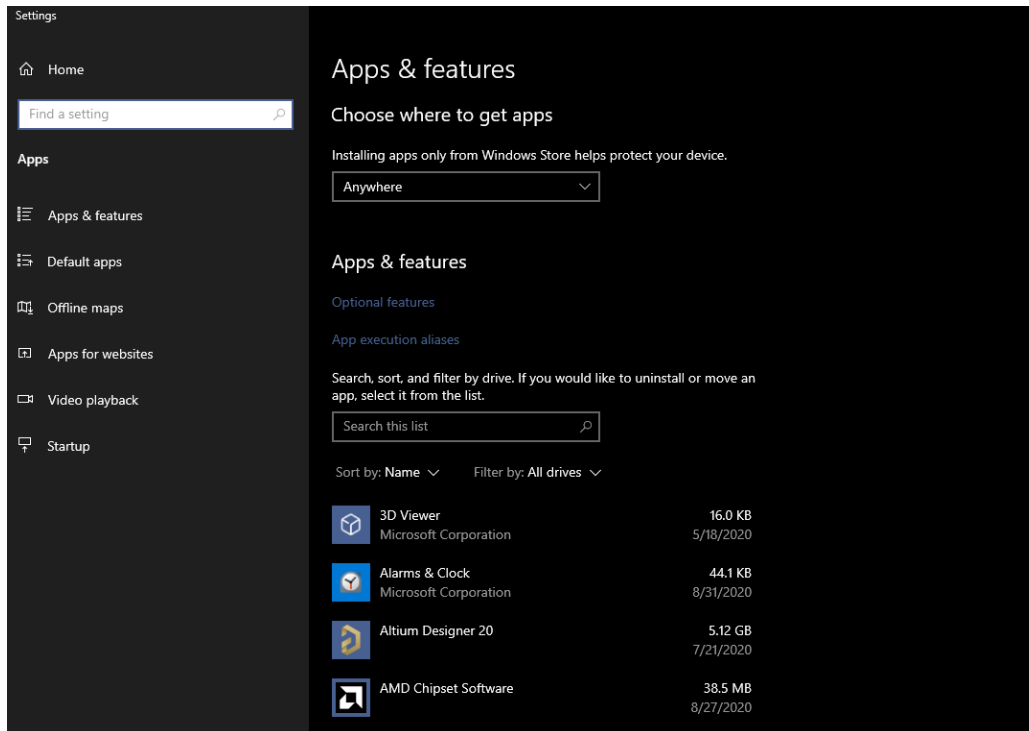
Step 9: Connect Raspberry pi to power
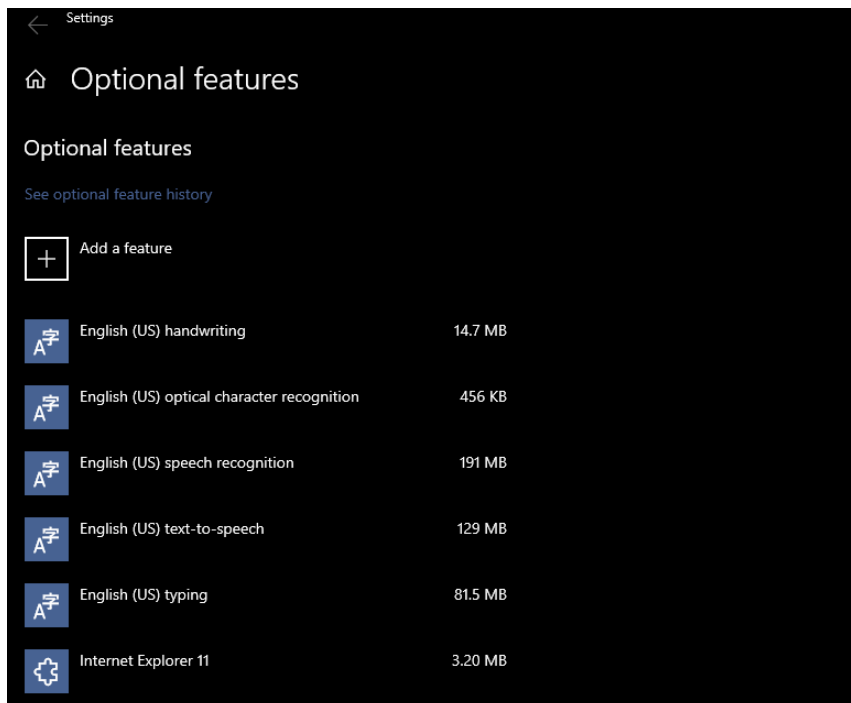
Step 10: Enable SSH in windows 10:

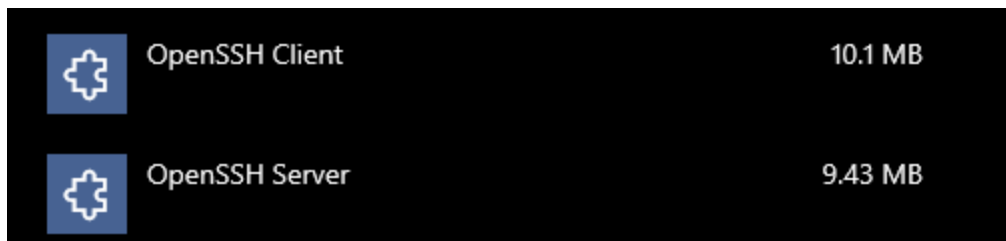1. Hit the windows key and type in "app" and then hit enter

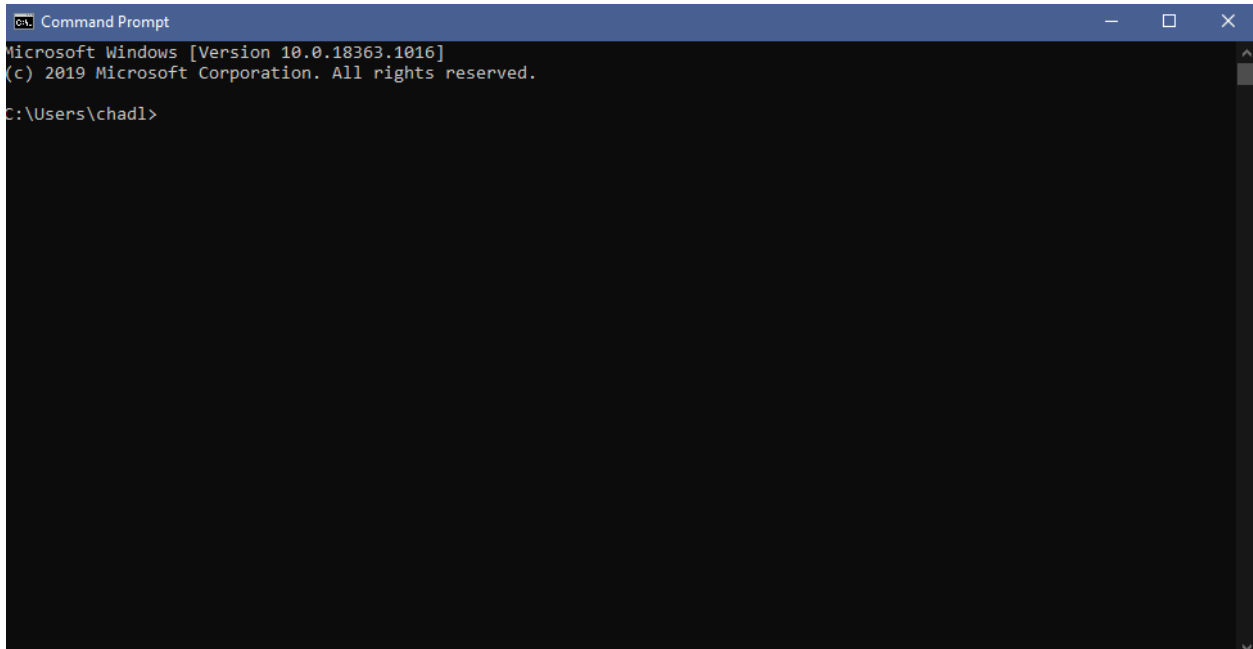2. Click on Optional features in the following page:

3. Click on "Add a feature"

4. Scroll down on the following page and find the following

⌂ Add a feature

Optional features

🧩	Arabic Script Supplemental Fonts	1.64 MB
🧩	Bangla Script Supplemental Fonts	482 KB
🧩	Canadian Aboriginal Syllabics Supplemental Fonts	96.6 KB
🧩	Cherokee Supplemental Fonts	76.7 KB
🧩	Chinese (Simplified) Supplemental Fonts	41.4 MB
🧩	Chinese (Traditional) Supplemental Fonts	12.8 MB
🧩	Devanagari Supplemental Fonts	1.47 MB
🧩	Ethiopic Supplemental Fonts	195 KB
🧩	Graphics Tools	15.0 MB
🧩	Gujarati Supplemental Fonts	212 KB
🧩	Gurmukhi Supplemental Fonts	114 KB
🧩	Hebrew Supplemental Fonts	377 KB
🧩	IrDA infrared	589 KB
🧩	Japanese Supplemental Fonts	50.6 MB
🧩	Kannada Supplemental Fonts	206 KB
🧩	Khmer Supplemental Fonts	413 KB
🧩	Korean Supplemental Fonts	9.30 MB
🧩	Lao Supplemental Fonts	190 KB
🧩	Malayalam Supplemental Fonts	166 KB
🧩	Microsoft WebDriver	321 KB

🧩	OpenSSH Client	10.1 MB
🧩	OpenSSH Server	9.43 MB

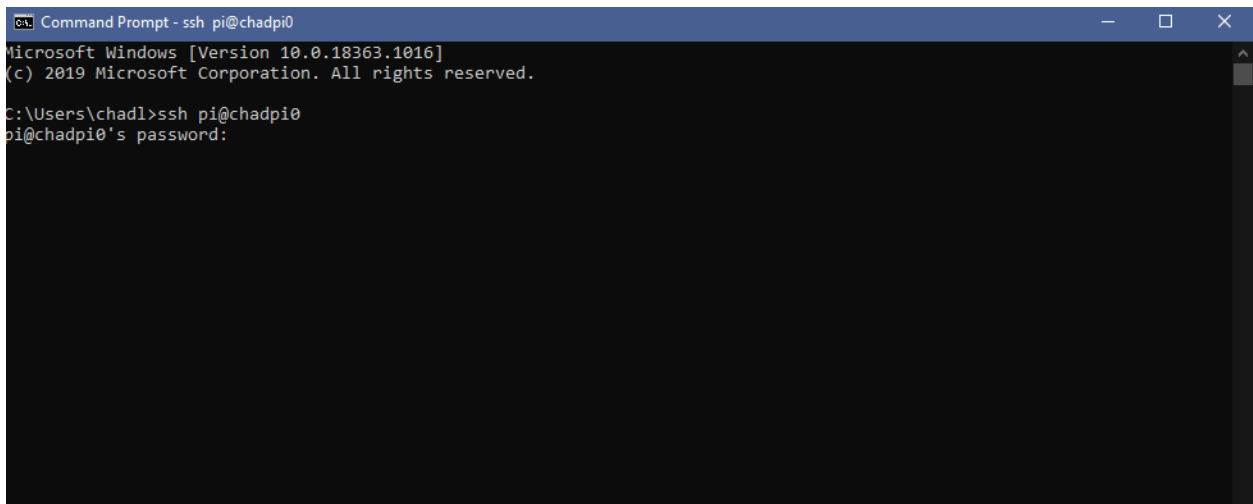5. Click and Install OpenSSH Client and OpenSSH Server

Step 11: **FOR WIFI SSH:**

1. Plug in raspberry pi into power
2. Open Command Prompt in windows 10 on computer you want to control the raspberry pi with

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\chadl>
```
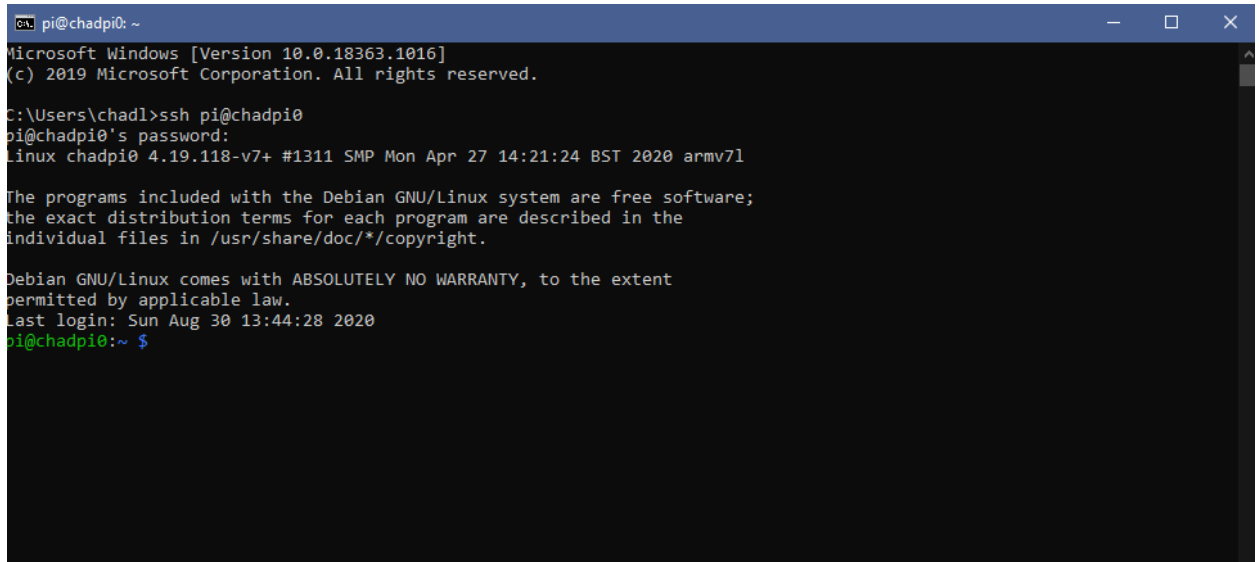
3. **Type ssh pi@"pi"**

```
Command Prompt - ssh  pi@chadpi0
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\chadl>ssh pi@chadpi0
pi@chadpi0's password:
```

4. Type in password made for raspberry pi **(DEFAULT [PASSWORD IS *raspberry*)**

Step 12: Complete WiFi SSH

Step 13: Ethernet - same as wifi steps, just plug ethernet into router

Step 14: Ethernet LAN - same as Ethernet but plug into computer that you will be controlling the pi with

COMPLETE!!!

To set up your first python code document:

Step1: Enter the following lines of code:
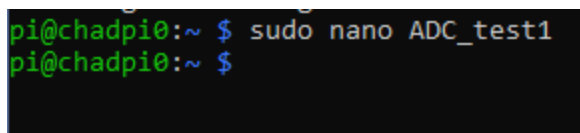
sudo echo filename

Sudo echo creates a text file with whatever name you specify (filename)

sudo nano filename

Sudo nano creates a text file that you created

sudo python3 filename

sudo python3 runs the text as python code

```python
#MCP3008 ADC Conversion
import time
import busio
import digitalio
import board
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

# create the spi bus
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)
cs = digitalio.DigitalInOut(board.D5)

# create the mcp object
mcp = MCP.MCP3008(spi, cs)

while True:
        # create an analog input channel on pin 0
        chan = AnalogIn(mcp, MCP.P0)
        temp = str((((chan.voltage*1000) - 500)/10))
        tempF = str((((((chan.voltage*1000) - 500)/10)*9/5) + 32)
        print('Raw ADC Value: ', chan.value)
        print('ADC Voltage: ' + str(chan.voltage) + 'V')
        print('it is : ' + (temp) + ' C or \n' + tempF + ' F')
                                          [ Read 26 lines ]
```

Hit CTRL + X to exit nano

If you edit nano text, hit CTRL + X, Y, and then ENTER

To stop a continuous python3 progarm loop, hit CTRL + C:

```
C:\. pi@chadpi0: ~
85.86157015335317 F
Raw ADC Value:  15872
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  15872
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  14528
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  15872
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  15872
ADC Voltage: 0.7218860151064316V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  15872
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  14528
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  15808
ADC Voltage: 0.6993270771343556V
it is : 29.92309452964065 C or
85.86157015335317 F
```

After hittingCTRL+C:

```
it is : 29.92309452964065 C or
85.86157015335317 F
Raw ADC Value:  15872
ADC Voltage: 0.7992309452964065V
it is : 29.92309452964065 C or
85.86157015335317 F
^CTraceback (most recent call last):
  File "ADC_test1", line 26, in <module>
    time.sleep(1)
KeyboardInterrupt
pi@chadpi0:~ $
```

There you go!