```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Brandon Bryant
import sys,os,base64,binascii,hashlib
def main():
    wrapper_file,output_folder = '','' # We need to store our file for analysis and where we are going to output the file
    for arg in sys.argv:
        if arg[0:2] == '-w':
                wrapper_file = arg[2::]
        if arg[0:2] == '-o':
                output_folder = arg[2::]
    if (wrapper_file == '' or output_folder == ''):
        print "You must input a file to analysis and a folder name to output to"
        sys.exit() # exit if wrapper and output folder not specified
    decode(wrapper_file,output_folder)
def decode(w,f):
    filetypes = [] # initialize our filetype array
    jpeg = {'header':"ffd8ff",'footer':"ffd9","filetype":".jpeg"}
    jpg ={'header':"ffd8ffe000104a46494600010101004800480000ffe100804578696600004d4d002a000000080004011a0005000000010000003e011b0005000000010000004601280003000000010002000087690004000000010000004e00000000000000480000000100000048000000010003a00100030000000100010000a00200040000000100000018ba003000400000001000002000",'footer':"2a45007fffd9","filetype":".jpg"}
    pdf = {'header':'25504446','footer':'454f460a',"filetype":".pdf"}
    gif = {'header':'47494638','footer':'5a00003b',"filetype":".gif"}
    png = {'header':'89504e47','footer':'49454e44ae426082',"filetype":".png"}
    docx = {'header':'504b0304','footer':'504b0506',"filetype":".docx"}
    filetypes.extend([jpeg,pdf,gif,png,docx,jpg]) #add our filetypes to the filetype array
    if os.path.exists(w):
            input_file = open(w,"r") # open our wrapper if exists
        else:
            sys.exit("file doesn't exist")
    if not os.path.exists(f):
        os.makedirs(f) # create the output folder
    data = input_file.read() # read and store data to variable from wrapper
    decoded = base64.b64decode(data) #decode the file from base64
    decoded = binascii.hexlify(decoded) #convert the data to string hex format
    i,begin,end = 0,0,0 # initialize uesful variables
    found_begin,found_end = False,False # initialize uesful variables
    for filetype in filetypes: #loop through each filetype
        while(i < len(decoded)): #loop over the contents of the file
            if (decoded[i:i+(len(filetype["header"]))] == filetype["header"]) and not found_begin: #search for the header
                begin = i   #store the location of the first byte of the header
                found_begin = True #header is found
            if (decoded[i:i+(len(filetype["footer"]))] == filetype["footer"]) and not found_end: #search for the footer
                end = i+(len(filetype["footer"])) #search for the whole footer
                found_end = True #footer is found
                break #we dont need to continue
            i = i+1 #advance do the next byte (or half byte)
        if(filetype["filetype"]==".docx"): # .docx requires extra padding
            decoded_alt = binascii.unhexlify(decoded[begin:end+(2*18)])
        else:
            decoded_alt = binascii.unhexlify(decoded[begin:end]) #output the header->footer
        a = hashlib.md5() #here we are going to hash the file for namesake
        a.update(decoded_alt) #hash
        a_md5 = a.hexdigest() #hash
        file = open("./"+f+"/"+a_md5+filetype["filetype"],"w+") #create output file
        file.write(decoded_alt) # write to file
        file.close() #close file
        i,begin,end = 0,0,0
        found_begin,found_end = False,False
main()
```