```
###################################
# PROLOG SECTION
# Program name: LLaVa_describe_images.py
# Program function: Provide text description of images
# Target audience: Some Python programming experience
# and a general understanding of the Generative AI (GenAI)
# environment would be helpful.
# Written: 17 Feb 2025
# Updated: 7 Mar 2025
# By: Richard H Johnson and Anna van Raaphorst-Johnson
# Copyright: 2025 by VRJ Associates, LLC
#
# Description and Specifications:
#
# Using a directory containing images as input,
# ask the LLaVA multimodal model
# running in the LLamaIndex framework (version 0.11.21)
# on the ollama Python server (version 0.3.3)
# to describe a set of images
# and capture the responses in a text file.
#
# The directory containing the images is part of
# our "Computer History" knowledge base (#7 in the set).
#
# According to Perplexity,
# LLaVA (Large Language and Vision Assistant)
# is a cutting-edge multimodal AI model that integrates
# natural language processing with visual understanding.
###################################

###################################
# ENVIRONMENT SETUP SECTION
# Import: ollama
###################################

import ollama
import os

###################################
# FUNCTION DEFINITION SECTION
###################################

#
# Ask the model for an image description
#
def GetDescription(image_path):
    bimage = os.path.basename(image)
    print("\nProcessing image:",bimage)
    print("\nProcessing image:",bimage,"\n",file=fd)
```

```
    res = ollama.chat(
        model=model_name,
        options={"temperature": model_temperature},
        messages=[
            {'role':'system',
             'content': system_prompt
            },
            {'role':'user',
             'content':SetQuery(image_path),
             'images':[image_path]
            }
        ]
    )

    return res['message']['content']

#
# Create a query string for an image
#
def SetQuery(image_path):
    bimage = os.path.basename(image_path)
    prompt = "This image file name is: "+'"'+bimage+'". '
    prompt = prompt+"Using this name as a hint, Describe this image. "
    prompt = prompt+"Dont make up anything."
    #print("Prompt: ",prompt)
    return prompt

#
# Function to test if file is an image
#
def isImage(f):
    iexts = (".PNG",".JPG",".GIF",".BMP",".JPEG")
    fdata = os.path.splitext(f)
    fext = fdata[1]
    fext = fext.upper()
    if fext in iexts:
        return True
    else:
        return False

###################################
# PROCESSING INITIALIZATION SECTION
###################################

# lower the temperature from the 0.8 default
# to avoid "flights of fancy" in the descriptions
model_temperature = 0.4

system_prompt = """
You are a helpful assistant. Limit your responses to 50 words or
```

```
fewer.
"""

model_name = "llava"

datadir = "../7_COMPUTERhistory_PUBLISHED_22Jan2025_copy/
IMAGES_NewsFromNan_Metaslider_ComputerHistory_10Jan2025"

outfile = "llava_output_log.txt"

fd = open(outfile,'w')

###################################
# PROCESSING SECTION
###################################

print("Processing images in",datadir,"with",model_name)
print("   temperature =",model_temperature)
print("   system prompt=",system_prompt)

# make a list of the iaages in the specified directory
icount=0
dir_list = os.listdir(datadir)

if len(dir_list)==0:
    print("No files in",datadir)
    exit(0)

# loop through the image files
for image in sorted(dir_list):
    if isImage(image):
        image_path = datadir+"/"+image
        ret = GetDescription(image_path)
        print("\n"+ret)
        print("\n"+ret,file=fd)
        icount=icount+1

print()
print(icount,"images processed")

fd.close()
```