A PROJECT REPORT ON COMPUTER SHOP MANAGEMENT

FOR
AISSCE 2022 EXAMINATION
[AS A PART OF INFORMATICS PRACTISES COURSE (065)]

SUBMITTED BY:

Master Aditya Gaurav
12th Science
Roll No. 25

UNDER THE GUIDANCE OF

Mrs Juliet Belly

CERTIFICATE ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to our Principal Sir **Mr. Jacob Thomas** as well as my teacher **Mrs. Juliet Belly** who gave me the golden opportunity to do this wonderful project on the topic **'Computer Shop Chatbot'**, which also helped me in doing a lot of research. I came to know about so many new things. I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within a limited time frame.

ABSTRACT

Project **'Computer Shop Chatbot'** describes an application on managing the important details of computer delicate & costly parts on a large scale, using efficient division of their properties into a managed solution, made in this project. This is useful for B2B as well as D2C chains.

CONTENTS

INTRODUCTION

The project titled **'Computer Shop Chatbot'** is an application of Python, Python Pandas, Matplotlib, and CSV files. This application is designed and coded in **'IDLE Python'** downloaded in the system. Python Pandas – Series (one-dimensional) and DataFrame (two-dimensional) are also used for more efficiency of the project.

The purpose of the project is to help people to understand more about CPUs, GPUs and related things, and help them get clarity to buy according to their needs.

Such types of projects have wider scope in the future and can be useful to many. This project is made in python and therefore it helps in understanding the basic concepts of Python, Pandas, Matplotlib, and CSV files. Hence, performing such programs helps to increase my knowledge of the subject. This knowledge is very important in my life in this 21st century which promotes IT and Technology.

- **AIM**

To make a project on 'Computer Shop Chatbot'' as an application of Python, Pandas, Matplotlib, and CSV files.

- **OBJECTIVE**

The main objective of this project is to ease the work & research of the consumers and shopkeepers. It reduces the shopkeeper's hard work manually of telling details of components by himself. Instead, now, anybody can use this chatbot to get all information about components. Moreover, it will help them keep an accurate record of all the details of each component. It enables the consumers to access, update and delete important details of CPUs and GPUs like their Clock Speeds, VRAMs, Performance Benchmarks and Date of launch record. It also has graphs depicting the components benchmarks and the Leaderboards' progress.

The purpose of the project is that the users (here Shopkeepers) can easily enter and access the details of components by chatting with the chatbot with proper accuracy. This can stunningly ease the work of the shopkeepers & consumers in recording & getting all the details accurately.

REQUIREMENT ANALYSIS:

TECHNOLOGIES USED:

Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. This programming language lets you work quickly and integrate systems more efficiently. Because of its simplicity in coding and powerful features, it works brilliantly for both beginners and experts. It's easy to use and is free. It's useful for a number of applications, including education, data analytics, and web development.

Python Pandas:

It is the most famous Python package for data science, which offers powerful and flexible data structures that make data analysis and manipulation easy. It is a package useful for data analysis and manipulation. Pandas provide an easy way to create, manipulate and wrangle the data. Pandas provide powerful and easy-to-use data structures.

Series (one-dimensional)

DataFrame (two-dimensional)

CSV files:

A CSV is a comma-separated values file, which allows data to be saved in a tabular format. CSV is a simple file such as a spreadsheet or database. Files in the CSV format can be imported and exported from programs that store data in tables, such as Microsoft Excel or Open Office.

Matplotlib:

Matplotlib is a Python 2D plotting library that produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib tries to make easy things easy and hard things possible.

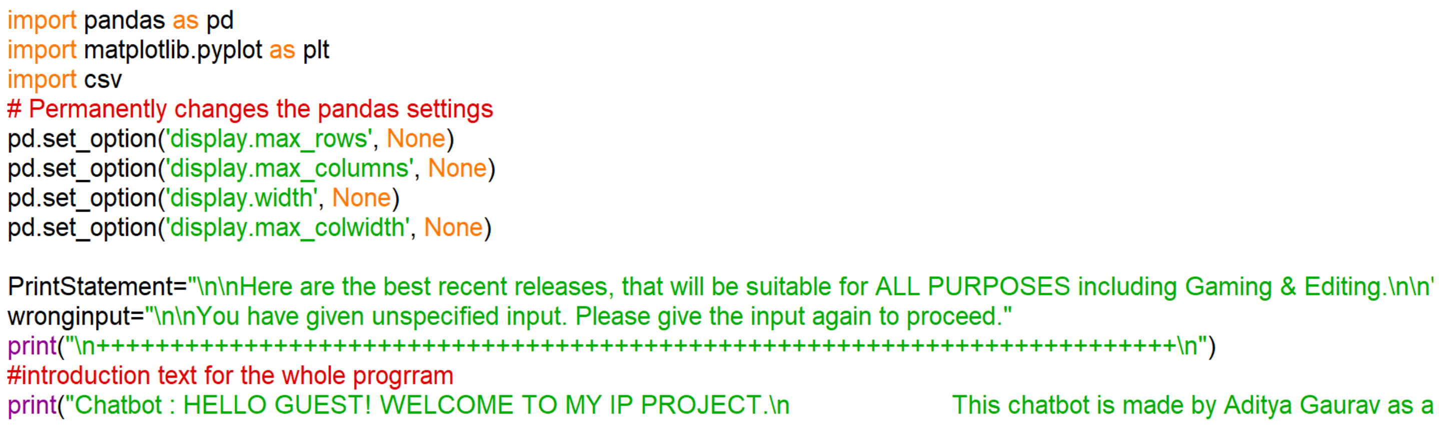
You can generate plots,
histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. It supports Interactive and non-interactive plotting and can save images in several output formats (PNG, PS & Others).

SOFTWARE REQUIREMENT:

1. IDLE (Python 3.8 32-bit)
2. Installed matplotlib.pyplot
3. Installed pandas
4. MS Word
5. Web browser: Google Chrome

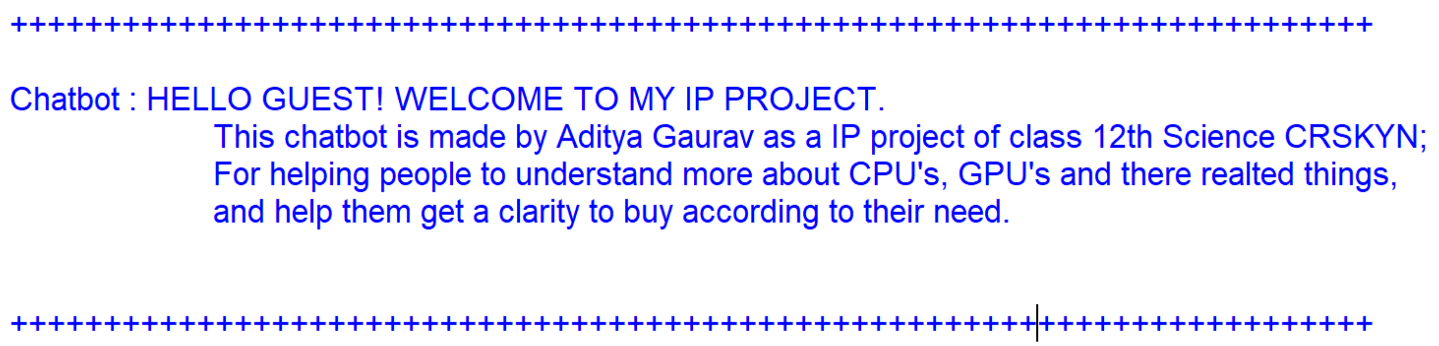
DESIGN AND CODING:

Welcome Coding:



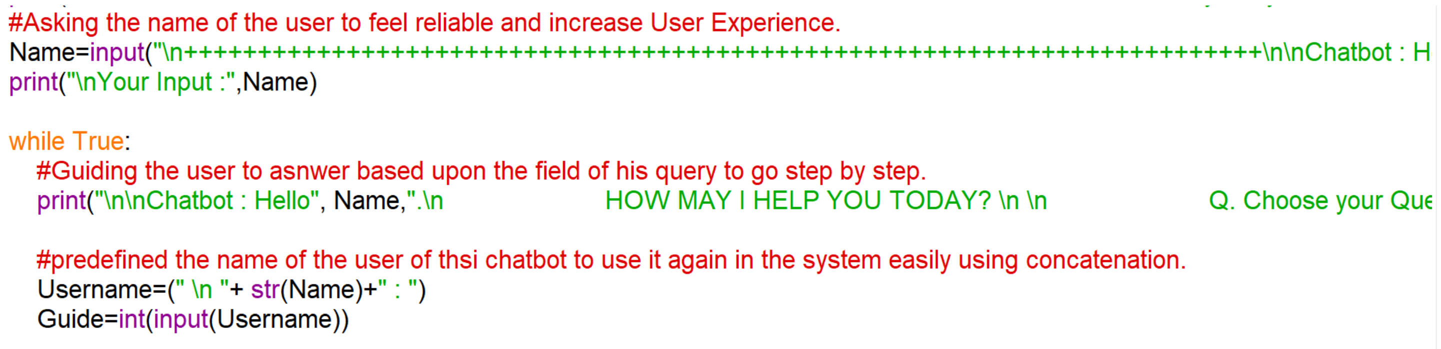
```
import pandas as pd
import matplotlib.pyplot as plt
import csv
# Permanently changes the pandas settings
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth', None)

PrintStatement="\n\nHere are the best recent releases, that will be suitable for ALL PURPOSES including Gaming & Editing.\n\n'
wronginput="\n\nYou have given unspecified input. Please give the input again to proceed."
print("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n")
#introduction text for the whole progrram
print("Chatbot : HELLO GUEST! WELCOME TO MY IP PROJECT.\n                    This chatbot is made by Aditya Gaurav as a
```

Output:



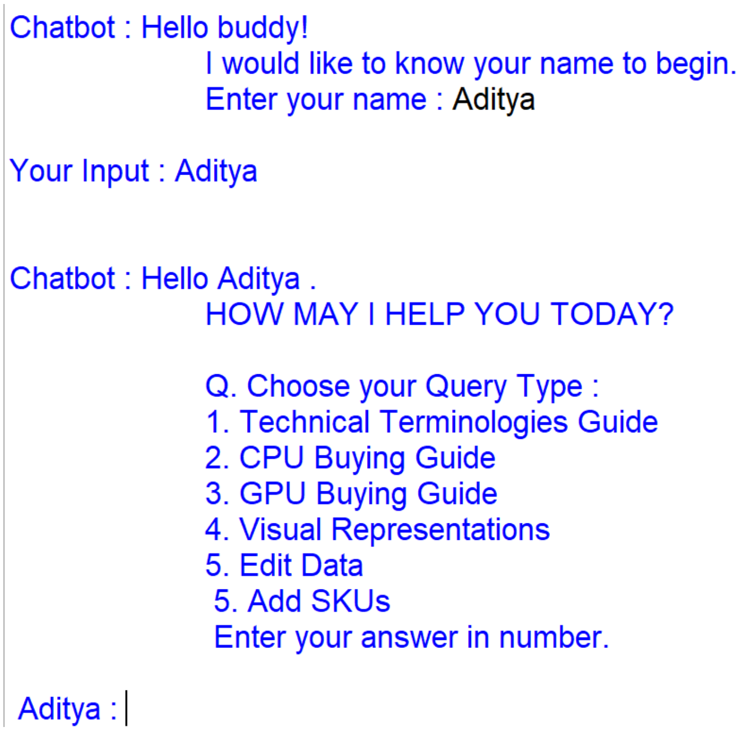
```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Chatbot : HELLO GUEST! WELCOME TO MY IP PROJECT.
                    This chatbot is made by Aditya Gaurav as a IP project of class 12th Science CRSKYN;
                    For helping people to understand more about CPU's, GPU's and there realted things,
                    and help them get a clarity to buy according to their need.


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Choice 1:



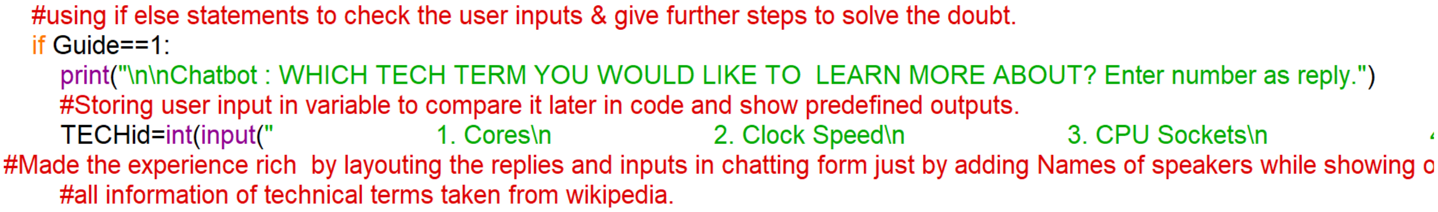
```
#Asking the name of the user to feel reliable and increase User Experience.
Name=input("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n\nChatbot : H
print("\nYour Input :",Name)

while True:
   #Guiding the user to asnwer based upon the field of his query to go step by step.
   print("\n\nChatbot : Hello", Name,".\n                    HOW MAY I HELP YOU TODAY? \n \n                    Q. Choose your Que

   #predefined the name of the user of thsi chatbot to use it again in the system easily using concatenation.
   Username=(" \n "+ str(Name)+" : ")
   Guide=int(input(Username))
```

Output:



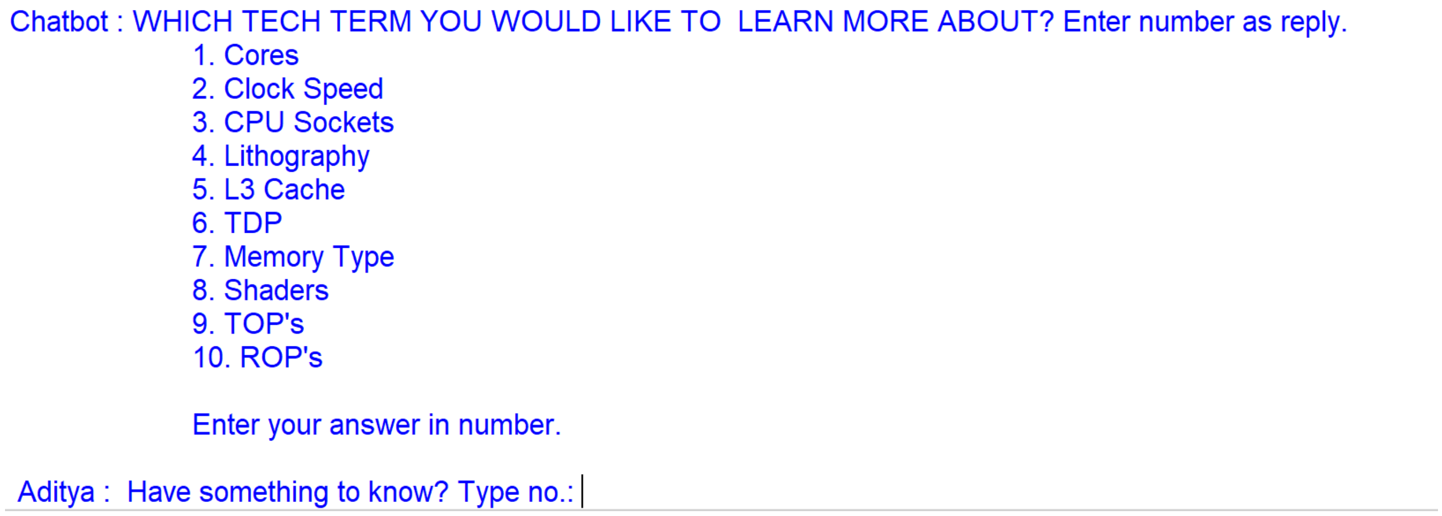
```
Chatbot : Hello buddy!
                    I would like to know your name to begin.
                    Enter your name : Aditya

Your Input : Aditya


Chatbot : Hello Aditya .
                    HOW MAY I HELP YOU TODAY?

                    Q. Choose your Query Type :
                    1. Technical Terminologies Guide
                    2. CPU Buying Guide
                    3. GPU Buying Guide
                    4. Visual Representations
                    5. Edit Data
                     5. Add SKUs
                     Enter your answer in number.

 Aditya : |
```

Inner Choice 1:



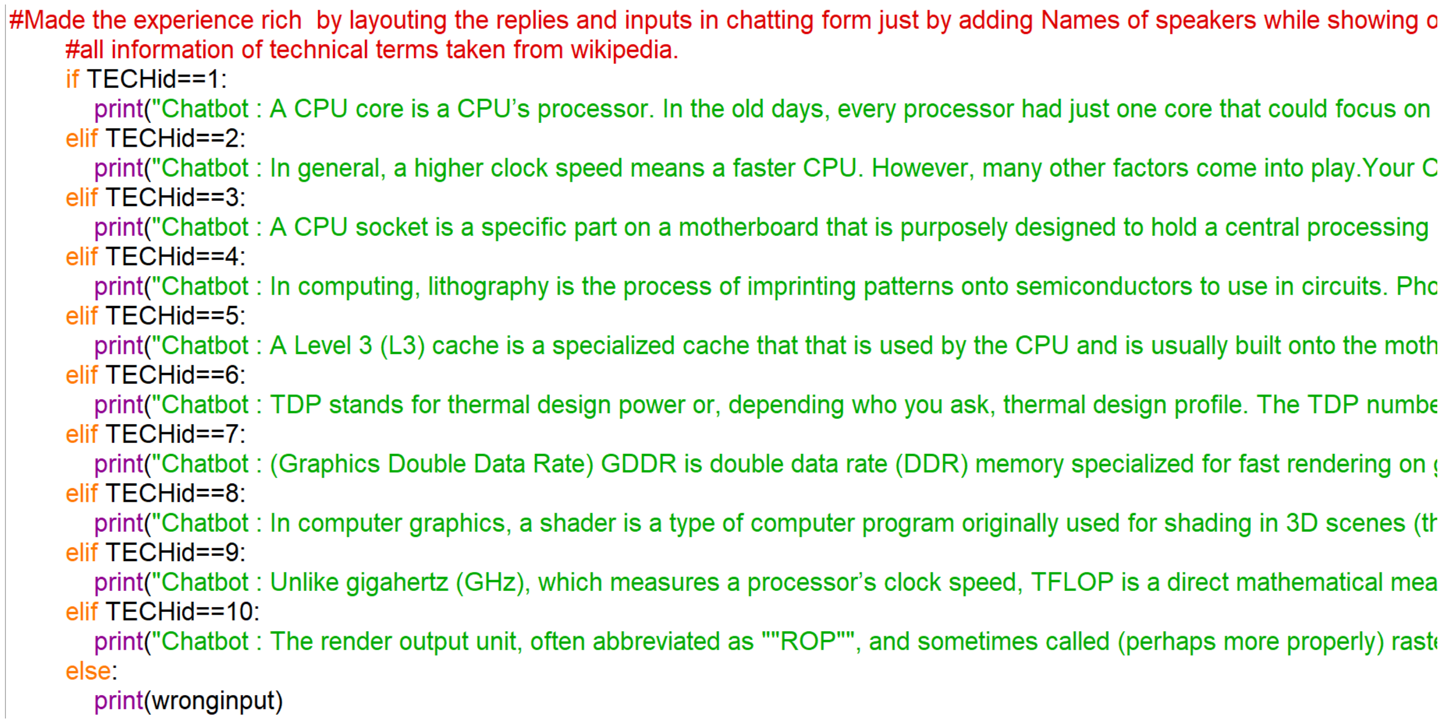
```
    #using if else statements to check the user inputs & give further steps to solve the doubt.
    if Guide==1:
        print("\n\nChatbot : WHICH TECH TERM YOU WOULD LIKE TO  LEARN MORE ABOUT? Enter number as reply.")
        #Storing user input in variable to compare it later in code and show predefined outputs.
        TECHid=int(input("                    1. Cores\n                    2. Clock Speed\n                    3. CPU Sockets\n
#Made the experience rich  by layouting the replies and inputs in chatting form just by adding Names of speakers while showing o
        #all information of technical terms taken from wikipedia.
```

Output:



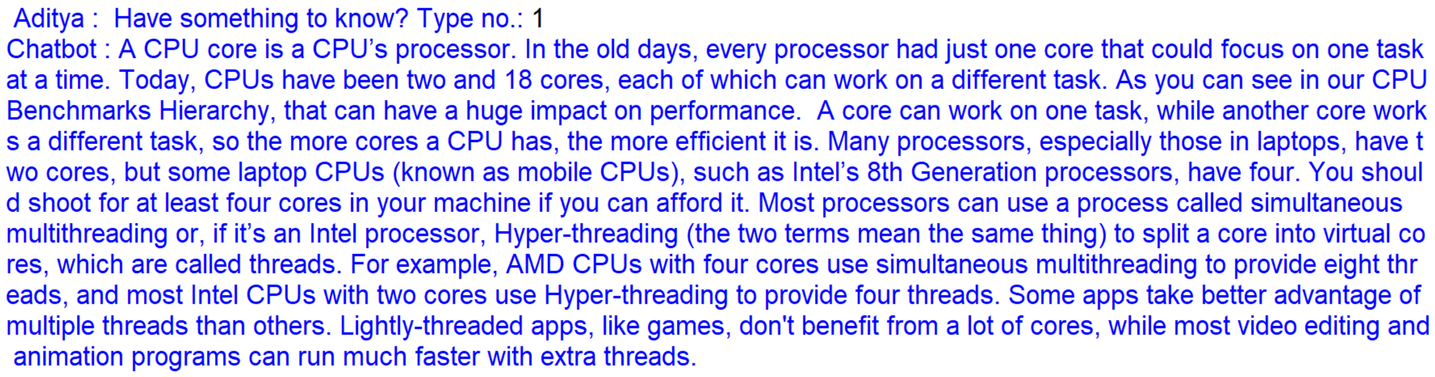
```
 Chatbot : WHICH TECH TERM YOU WOULD LIKE TO  LEARN MORE ABOUT? Enter number as reply.
                    1. Cores
                    2. Clock Speed
                    3. CPU Sockets
                    4. Lithography
                    5. L3 Cache
                    6. TDP
                    7. Memory Type
                    8. Shaders
                    9. TOP's
                    10. ROP's

                    Enter your answer in number.

  Aditya :  Have something to know? Type no.:
```

To learn about different key terms:

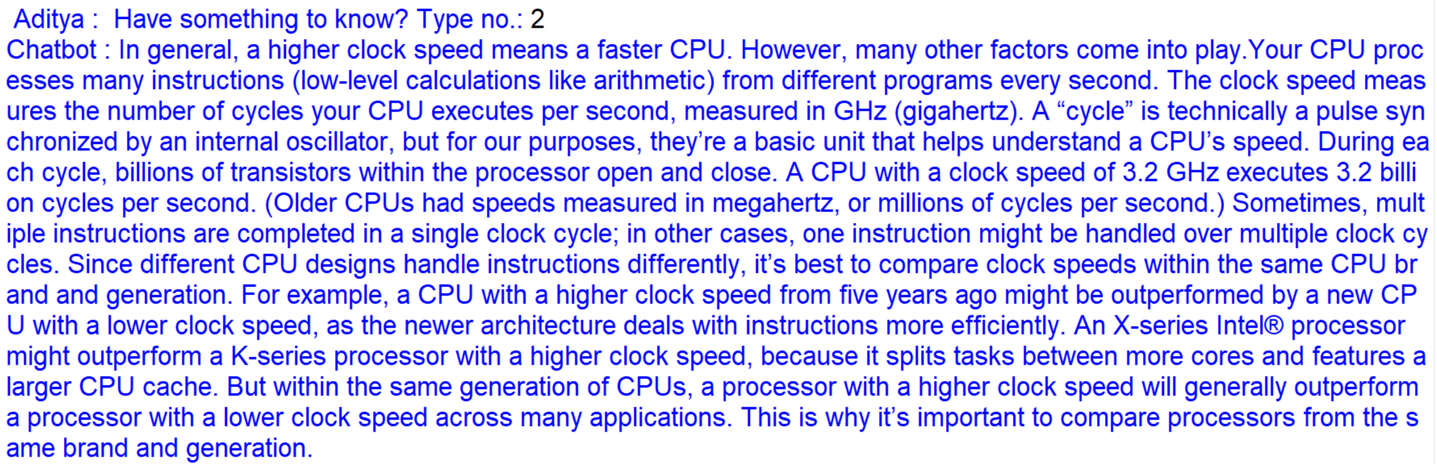


```
#Made the experience rich  by layouting the replies and inputs in chatting form just by adding Names of speakers while showing o
    #all information of technical terms taken from wikipedia.
    if TECHid==1:
      print("Chatbot : A CPU core is a CPU’s processor. In the old days, every processor had just one core that could focus on
    elif TECHid==2:
      print("Chatbot : In general, a higher clock speed means a faster CPU. However, many other factors come into play.Your C
    elif TECHid==3:
      print("Chatbot : A CPU socket is a specific part on a motherboard that is purposely designed to hold a central processing
    elif TECHid==4:
      print("Chatbot : In computing, lithography is the process of imprinting patterns onto semiconductors to use in circuits. Pho
    elif TECHid==5:
      print("Chatbot : A Level 3 (L3) cache is a specialized cache that that is used by the CPU and is usually built onto the moth
    elif TECHid==6:
      print("Chatbot : TDP stands for thermal design power or, depending who you ask, thermal design profile. The TDP numbe
    elif TECHid==7:
      print("Chatbot : (Graphics Double Data Rate) GDDR is double data rate (DDR) memory specialized for fast rendering on g
    elif TECHid==8:
      print("Chatbot : In computer graphics, a shader is a type of computer program originally used for shading in 3D scenes (th
    elif TECHid==9:
      print("Chatbot : Unlike gigahertz (GHz), which measures a processor’s clock speed, TFLOP is a direct mathematical mea
    elif TECHid==10:
      print("Chatbot : The render output unit, often abbreviated as ""ROP"", and sometimes called (perhaps more properly) raste
    else:
      print(wronginput)
```

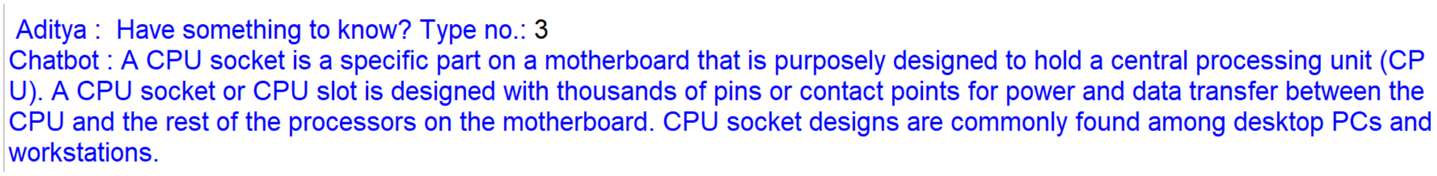
Outputs:



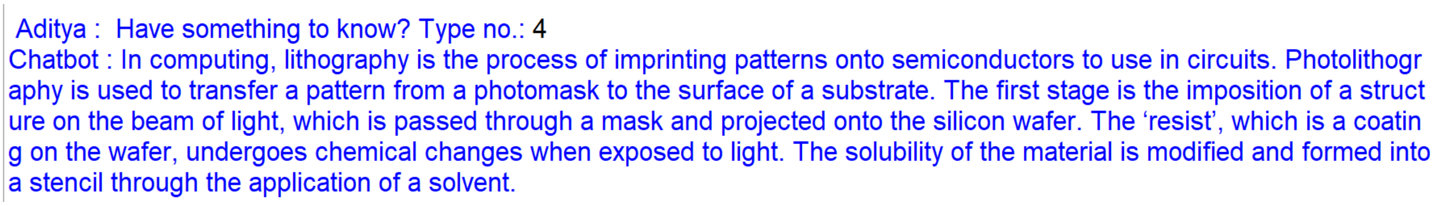
```
 Aditya :  Have something to know? Type no.: 1
Chatbot : A CPU core is a CPU’s processor. In the old days, every processor had just one core that could focus on one task
at a time. Today, CPUs have been two and 18 cores, each of which can work on a different task. As you can see in our CPU
Benchmarks Hierarchy, that can have a huge impact on performance.  A core can work on one task, while another core work
s a different task, so the more cores a CPU has, the more efficient it is. Many processors, especially those in laptops, have t
wo cores, but some laptop CPUs (known as mobile CPUs), such as Intel’s 8th Generation processors, have four. You shoul
d shoot for at least four cores in your machine if you can afford it. Most processors can use a process called simultaneous
multithreading or, if it’s an Intel processor, Hyper-threading (the two terms mean the same thing) to split a core into virtual co
res, which are called threads. For example, AMD CPUs with four cores use simultaneous multithreading to provide eight thr
eads, and most Intel CPUs with two cores use Hyper-threading to provide four threads. Some apps take better advantage of
multiple threads than others. Lightly-threaded apps, like games, don't benefit from a lot of cores, while most video editing and
 animation programs can run much faster with extra threads.
```



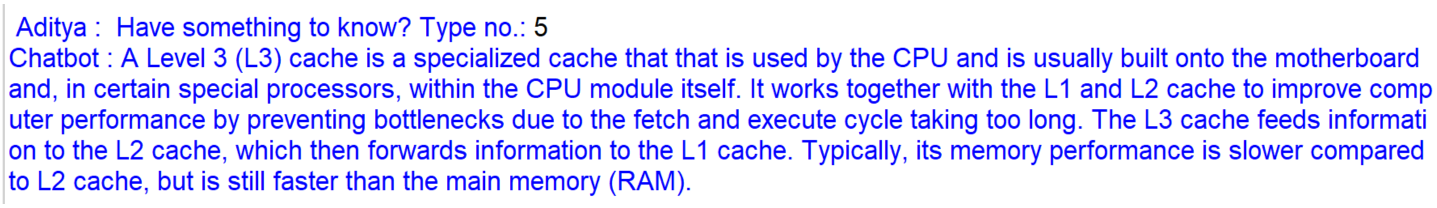
```
 Aditya :  Have something to know? Type no.: 2
Chatbot : In general, a higher clock speed means a faster CPU. However, many other factors come into play.Your CPU proc
esses many instructions (low-level calculations like arithmetic) from different programs every second. The clock speed meas
ures the number of cycles your CPU executes per second, measured in GHz (gigahertz). A “cycle” is technically a pulse syn
chronized by an internal oscillator, but for our purposes, they’re a basic unit that helps understand a CPU’s speed. During ea
ch cycle, billions of transistors within the processor open and close. A CPU with a clock speed of 3.2 GHz executes 3.2 billi
on cycles per second. (Older CPUs had speeds measured in megahertz, or millions of cycles per second.) Sometimes, mult
iple instructions are completed in a single clock cycle; in other cases, one instruction might be handled over multiple clock cy
cles. Since different CPU designs handle instructions differently, it’s best to compare clock speeds within the same CPU br
and and generation. For example, a CPU with a higher clock speed from five years ago might be outperformed by a new CP
U with a lower clock speed, as the newer architecture deals with instructions more efficiently. An X-series Intel® processor
might outperform a K-series processor with a higher clock speed, because it splits tasks between more cores and features a
larger CPU cache. But within the same generation of CPUs, a processor with a higher clock speed will generally outperform
a processor with a lower clock speed across many applications. This is why it’s important to compare processors from the s
ame brand and generation.
```



```
 Aditya :  Have something to know? Type no.: 3
Chatbot : A CPU socket is a specific part on a motherboard that is purposely designed to hold a central processing unit (CP
U). A CPU socket or CPU slot is designed with thousands of pins or contact points for power and data transfer between the
CPU and the rest of the processors on the motherboard. CPU socket designs are commonly found among desktop PCs and
workstations.
```



```
 Aditya :  Have something to know? Type no.: 4
Chatbot : In computing, lithography is the process of imprinting patterns onto semiconductors to use in circuits. Photolithogr
aphy is used to transfer a pattern from a photomask to the surface of a substrate. The first stage is the imposition of a struct
ure on the beam of light, which is passed through a mask and projected onto the silicon wafer. The ‘resist’, which is a coatin
g on the wafer, undergoes chemical changes when exposed to light. The solubility of the material is modified and formed into
a stencil through the application of a solvent.
```



```
 Aditya :  Have something to know? Type no.: 5
Chatbot : A Level 3 (L3) cache is a specialized cache that that is used by the CPU and is usually built onto the motherboard
and, in certain special processors, within the CPU module itself. It works together with the L1 and L2 cache to improve comp
uter performance by preventing bottlenecks due to the fetch and execute cycle taking too long. The L3 cache feeds informati
on to the L2 cache, which then forwards information to the L1 cache. Typically, its memory performance is slower compared
to L2 cache, but is still faster than the main memory (RAM).
```

Aditya : Have something to know? Type no.: 6

Chatbot : TDP stands for thermal design power or, depending who you ask, thermal design profile. The TDP number tells you the maximum heat a computer chip, such as a CPU or GPU, can use in watts. It also is often used as a basic indicator of power consumption. More watts equals better performance, but also higher temperatures and more power consumption. PC OEMs can sometimes lower the TDP on a chip to increase battery life on a laptop or raise it up to boost speed.

Aditya : Have something to know? Type no.: 7

Chatbot : (Graphics Double Data Rate) GDDR is double data rate (DDR) memory specialized for fast rendering on graphics cards (GPUs). Introduced in 2000, GDDR is the primary graphics RAM in use today. GDDR is technically GDDR SDRAM and supersedes VRAM and WRAM.

Each GDDR generation is faster and includes enhancements; however, although based on DDR memory, GDDR versions do not correspond numerically to DDR. For example, GDDR3 was based on DDR2 chips; GDDR5 on DDR3 and so on.

GDDR6 As of 2020, GDDR6 is the current version providing a 16 Gbps data rate instead of 13 Gbps for GDDR5X and twice that of GDDR5. GDDR6 also supports the same 32-byte read/write access as GDDR5. See GPU, DDR and SGRAM.

Aditya : Have something to know? Type no.: 8

Chatbot : In computer graphics, a shader is a type of computer program originally used for shading in 3D scenes (the production of appropriate levels of light, darkness, and color in a rendered image). They now perform a variety of specialized functions in various fields within the category of computer graphics special effects, or else do video post-processing unrelated to shading, or even perform functions unrelated to graphics.

Traditional shaders calculate rendering effects on graphics hardware with a high degree of flexibility. Most shaders are coded for (and run on) a graphics processing unit (GPU),[1] though this is not a strict requirement. Shading languages are used to program the GPU's rendering pipeline, which has mostly superseded the fixed-function pipeline of the past that only allowed for common geometry transforming and pixel-shading functions; with shaders, customized effects can be used. The position and color (hue, saturation, brightness, and contrast) of all pixels, vertices, and/or textures used to construct a final rendered image can be altered using algorithms defined in a shader, and can be modified by external variables or textures introduced by the computer program calling the shader.

Aditya : Have something to know? Type no.: 9

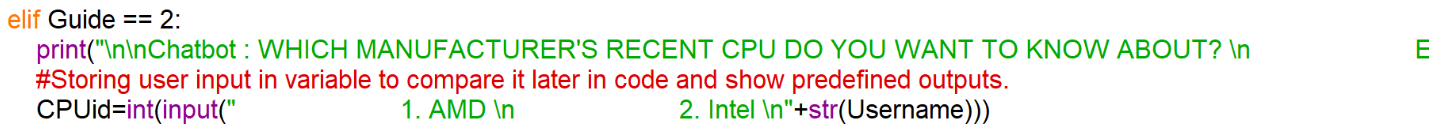
Chatbot : Unlike gigahertz (GHz), which measures a processor’s clock speed, TFLOP is a direct mathematical measurement of a computer’s performance.

Specifically, a teraflop refers to a processor’s capability to calculate one trillion floating-point operations per second. Saying something has “6 TFLOPS,” for example, means that its processor setup can handle 6 trillion floating-point calculations every second, on average.

Aditya : Have something to know? Type no.: 10

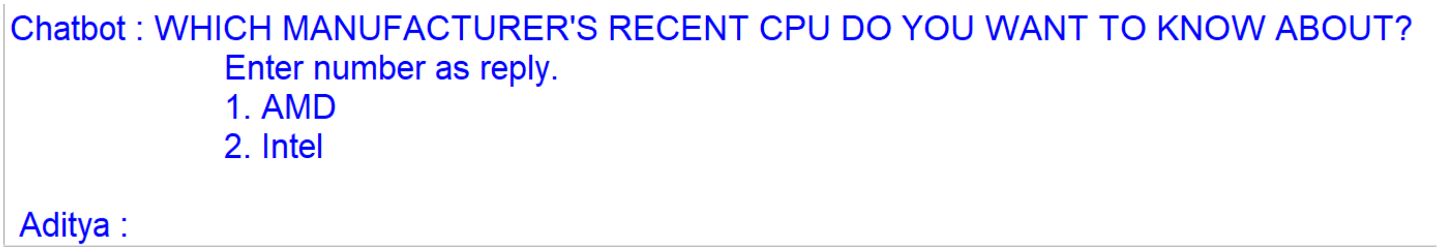
Chatbot : The render output unit, often abbreviated as ROP, and sometimes called (perhaps more properly) raster operations pipeline, is a hardware component in modern graphics processing units (GPUs) and one of the final steps in the rendering process of modern 3D accelerator boards. The pixel pipelines take pixel and texel information and process it, via specific matrix and vector operations, into a final pixel or depth value. This process is called rasterization. The ROPs perform the transactions between the relevant buffers in the local memory – this includes writing or reading values, as well as blending them together. Dedicated antialiasing hardware used to perform hardware-based antialiasing methods like MSAA is contained in ROPs.

2nd Choice - Choosing which CPU Brand to know more about:



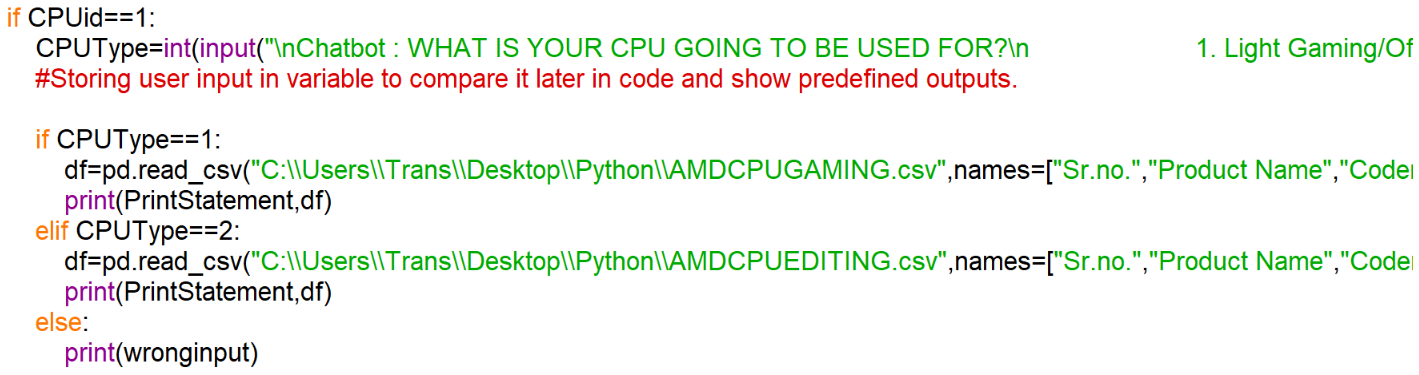
```
elif Guide == 2:
    print("\n\nChatbot : WHICH MANUFACTURER'S RECENT CPU DO YOU WANT TO KNOW ABOUT? \n                    E
    #Storing user input in variable to compare it later in code and show predefined outputs.
    CPUid=int(input("                    1. AMD \n                    2. Intel \n"+str(Username)))
```

Output:



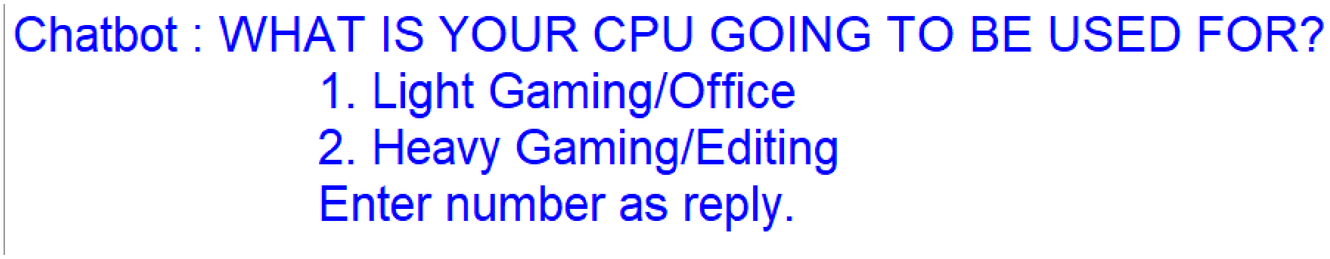
```
Chatbot : WHICH MANUFACTURER'S RECENT CPU DO YOU WANT TO KNOW ABOUT?
                    Enter number as reply.
                    1. AMD
                    2. Intel

 Aditya :
```

2nd Choice - To get the Latest CPUs of AMD as output:

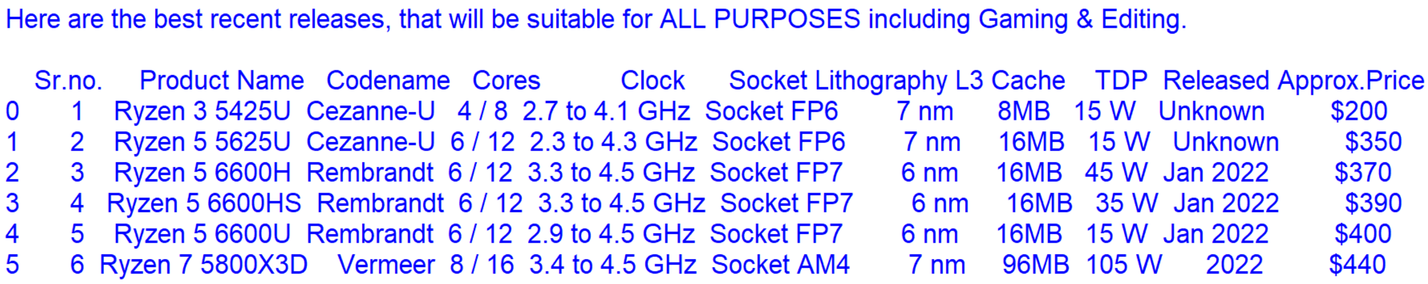


```
if CPUid==1:
    CPUType=int(input("\nChatbot : WHAT IS YOUR CPU GOING TO BE USED FOR?\n                    1. Light Gaming/Of
    #Storing user input in variable to compare it later in code and show predefined outputs.

    if CPUType==1:
        df=pd.read_csv("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUGAMING.csv",names=["Sr.no.","Product Name","Code
        print(PrintStatement,df)
    elif CPUType==2:
        df=pd.read_csv("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUEDITING.csv",names=["Sr.no.","Product Name","Code
        print(PrintStatement,df)
    else:
        print(wronginput)
```

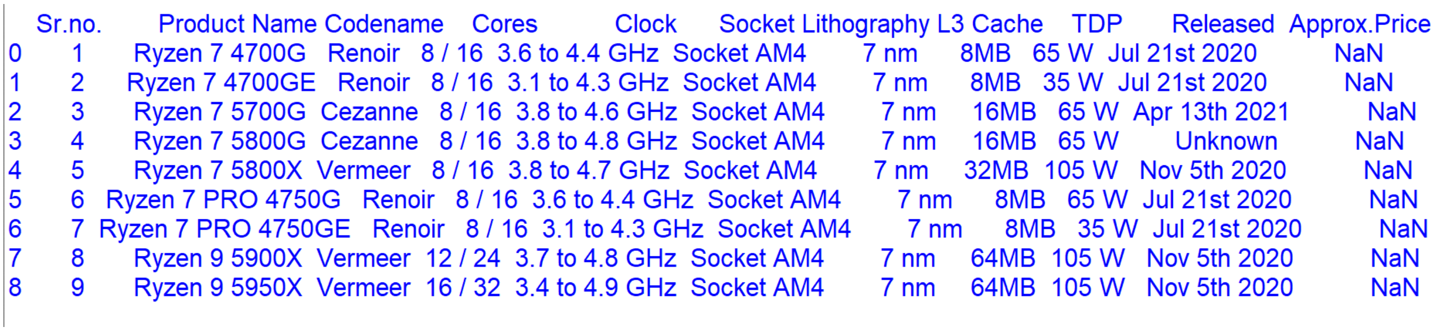
Output:



```
Chatbot : WHAT IS YOUR CPU GOING TO BE USED FOR?
                        1. Light Gaming/Office
                        2. Heavy Gaming/Editing
                        Enter number as reply.
```

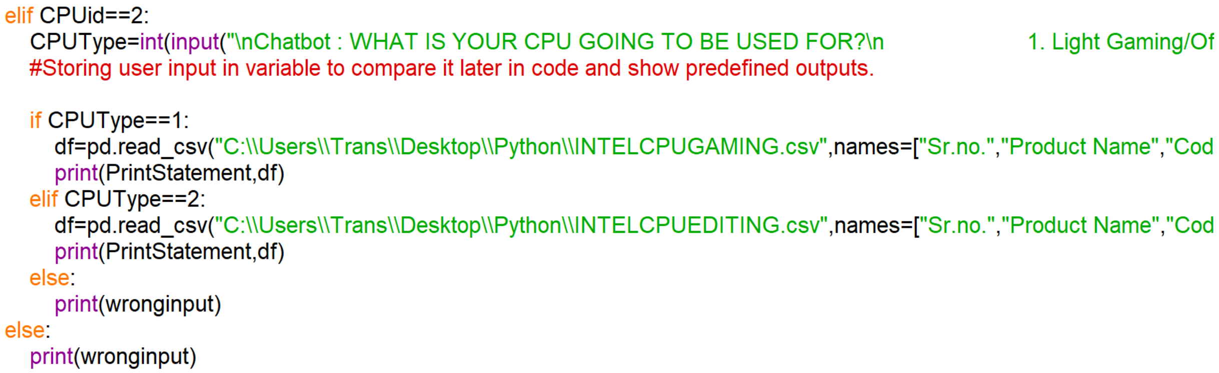


```
Here are the best recent releases, that will be suitable for ALL PURPOSES including Gaming & Editing.

   Sr.no.     Product Name   Codename   Cores        Clock     Socket Lithography L3 Cache   TDP  Released Approx.Price
0     1   Ryzen 3 5425U  Cezanne-U   4 / 8  2.7 to 4.1 GHz  Socket FP6       7 nm     8MB   15 W   Unknown        $200
1     2   Ryzen 5 5625U  Cezanne-U  6 / 12  2.3 to 4.3 GHz  Socket FP6       7 nm    16MB   15 W   Unknown        $350
2     3   Ryzen 5 6600H  Rembrandt  6 / 12  3.3 to 4.5 GHz  Socket FP7       6 nm    16MB   45 W  Jan 2022        $370
3     4  Ryzen 5 6600HS  Rembrandt  6 / 12  3.3 to 4.5 GHz  Socket FP7       6 nm    16MB   35 W  Jan 2022        $390
4     5   Ryzen 5 6600U  Rembrandt  6 / 12  2.9 to 4.5 GHz  Socket FP7       6 nm    16MB   15 W  Jan 2022        $400
5     6  Ryzen 7 5800X3D    Vermeer  8 / 16  3.4 to 4.5 GHz  Socket AM4       7 nm    96MB  105 W      2022        $440
```



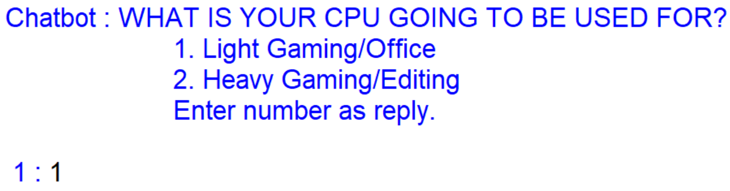
```
   Sr.no.       Product Name Codename    Cores        Clock     Socket Lithography L3 Cache    TDP       Released  Approx.Price
0     1      Ryzen 7 4700G   Renoir   8 / 16  3.6 to 4.4 GHz  Socket AM4       7 nm     8MB   65 W  Jul 21st 2020        NaN
1     2     Ryzen 7 4700GE   Renoir   8 / 16  3.1 to 4.3 GHz  Socket AM4       7 nm     8MB   35 W  Jul 21st 2020        NaN
2     3      Ryzen 7 5700G  Cezanne   8 / 16  3.8 to 4.6 GHz  Socket AM4       7 nm    16MB   65 W  Apr 13th 2021        NaN
3     4      Ryzen 7 5800G  Cezanne   8 / 16  3.8 to 4.8 GHz  Socket AM4       7 nm    16MB   65 W        Unknown        NaN
4     5      Ryzen 7 5800X  Vermeer   8 / 16  3.8 to 4.7 GHz  Socket AM4       7 nm    32MB  105 W   Nov 5th 2020        NaN
5     6  Ryzen 7 PRO 4750G   Renoir   8 / 16  3.6 to 4.4 GHz  Socket AM4       7 nm     8MB   65 W  Jul 21st 2020        NaN
6     7 Ryzen 7 PRO 4750GE   Renoir   8 / 16  3.1 to 4.3 GHz  Socket AM4       7 nm     8MB   35 W  Jul 21st 2020        NaN
7     8      Ryzen 9 5900X  Vermeer  12 / 24  3.7 to 4.8 GHz  Socket AM4       7 nm    64MB  105 W   Nov 5th 2020        NaN
8     9      Ryzen 9 5950X  Vermeer  16 / 32  3.4 to 4.9 GHz  Socket AM4       7 nm    64MB  105 W   Nov 5th 2020        NaN
```

2nd Choice -To get the Latest CPUs of Intel as output:

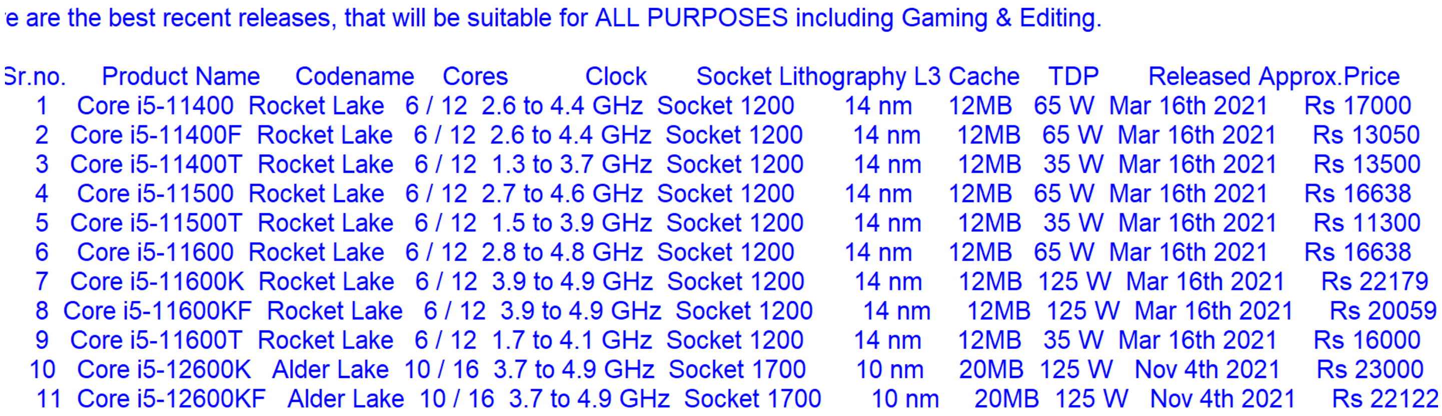


```
elif CPUid==2:
    CPUType=int(input("\nChatbot : WHAT IS YOUR CPU GOING TO BE USED FOR?\n                    1. Light Gaming/Of
    #Storing user input in variable to compare it later in code and show predefined outputs.

    if CPUType==1:
        df=pd.read_csv("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUGAMING.csv",names=["Sr.no.","Product Name","Cod
        print(PrintStatement,df)
    elif CPUType==2:
        df=pd.read_csv("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUEDITING.csv",names=["Sr.no.","Product Name","Cod
        print(PrintStatement,df)
    else:
        print(wronginput)
else:
    print(wronginput)
```

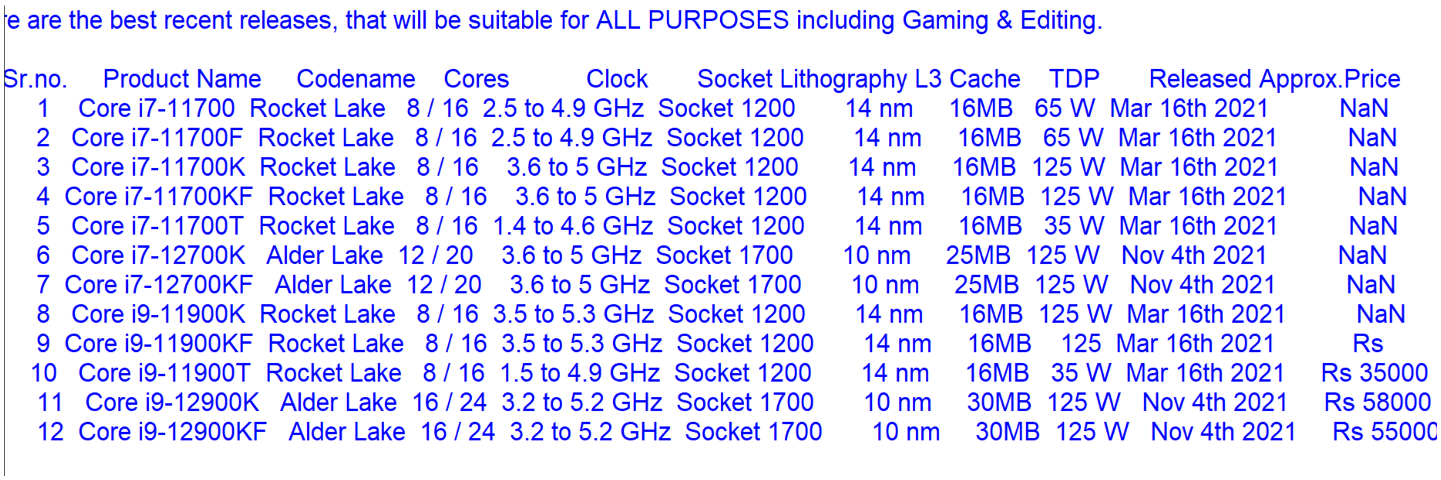
Output:



```
Chatbot : WHAT IS YOUR CPU GOING TO BE USED FOR?
                    1. Light Gaming/Office
                    2. Heavy Gaming/Editing
                    Enter number as reply.

 1 : 1
```

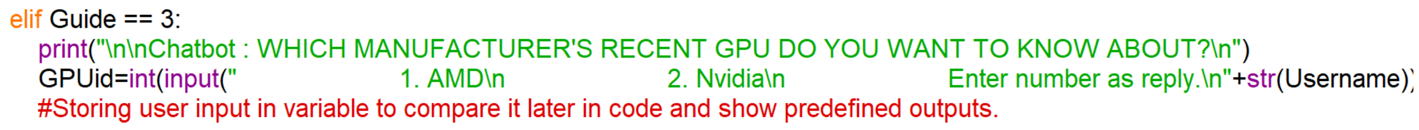


```
e are the best recent releases, that will be suitable for ALL PURPOSES including Gaming & Editing.

Sr.no.    Product Name    Codename    Cores        Clock      Socket Lithography L3 Cache    TDP      Released Approx.Price
   1    Core i5-11400  Rocket Lake   6 / 12  2.6 to 4.4 GHz  Socket 1200       14 nm     12MB   65 W  Mar 16th 2021     Rs 17000
   2   Core i5-11400F  Rocket Lake   6 / 12  2.6 to 4.4 GHz  Socket 1200       14 nm     12MB   65 W  Mar 16th 2021     Rs 13050
   3   Core i5-11400T  Rocket Lake   6 / 12  1.3 to 3.7 GHz  Socket 1200       14 nm     12MB   35 W  Mar 16th 2021     Rs 13500
   4    Core i5-11500  Rocket Lake   6 / 12  2.7 to 4.6 GHz  Socket 1200       14 nm     12MB   65 W  Mar 16th 2021     Rs 16638
   5   Core i5-11500T  Rocket Lake   6 / 12  1.5 to 3.9 GHz  Socket 1200       14 nm     12MB   35 W  Mar 16th 2021     Rs 11300
   6    Core i5-11600  Rocket Lake   6 / 12  2.8 to 4.8 GHz  Socket 1200       14 nm     12MB   65 W  Mar 16th 2021     Rs 16638
   7   Core i5-11600K  Rocket Lake   6 / 12  3.9 to 4.9 GHz  Socket 1200       14 nm     12MB  125 W  Mar 16th 2021     Rs 22179
   8  Core i5-11600KF  Rocket Lake   6 / 12  3.9 to 4.9 GHz  Socket 1200       14 nm     12MB  125 W  Mar 16th 2021     Rs 20059
   9   Core i5-11600T  Rocket Lake   6 / 12  1.7 to 4.1 GHz  Socket 1200       14 nm     12MB   35 W  Mar 16th 2021     Rs 16000
  10   Core i5-12600K   Alder Lake  10 / 16  3.7 to 4.9 GHz  Socket 1700       10 nm     20MB  125 W   Nov 4th 2021     Rs 23000
  11  Core i5-12600KF   Alder Lake  10 / 16  3.7 to 4.9 GHz  Socket 1700       10 nm     20MB  125 W   Nov 4th 2021     Rs 22122
```



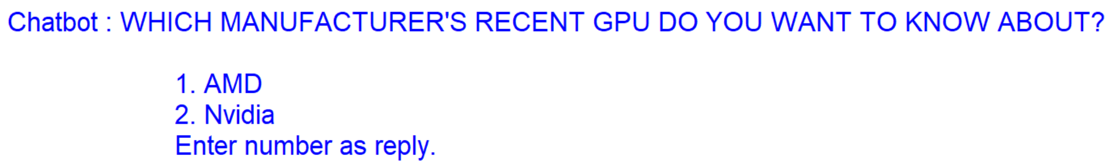
```
'e are the best recent releases, that will be suitable for ALL PURPOSES including Gaming & Editing.

Sr.no.    Product Name    Codename    Cores         Clock      Socket Lithography L3 Cache    TDP      Released Approx.Price
   1    Core i7-11700  Rocket Lake   8 / 16  2.5 to 4.9 GHz  Socket 1200      14 nm     16MB   65 W  Mar 16th 2021        NaN
   2   Core i7-11700F  Rocket Lake   8 / 16  2.5 to 4.9 GHz  Socket 1200      14 nm     16MB   65 W  Mar 16th 2021        NaN
   3   Core i7-11700K  Rocket Lake   8 / 16    3.6 to 5 GHz  Socket 1200      14 nm     16MB  125 W  Mar 16th 2021        NaN
   4  Core i7-11700KF  Rocket Lake   8 / 16    3.6 to 5 GHz  Socket 1200      14 nm     16MB  125 W  Mar 16th 2021        NaN
   5   Core i7-11700T  Rocket Lake   8 / 16  1.4 to 4.6 GHz  Socket 1200      14 nm     16MB   35 W  Mar 16th 2021        NaN
   6   Core i7-12700K   Alder Lake  12 / 20    3.6 to 5 GHz  Socket 1700      10 nm     25MB  125 W   Nov 4th 2021        NaN
   7  Core i7-12700KF   Alder Lake  12 / 20    3.6 to 5 GHz  Socket 1700      10 nm     25MB  125 W   Nov 4th 2021        NaN
   8   Core i9-11900K  Rocket Lake   8 / 16  3.5 to 5.3 GHz  Socket 1200      14 nm     16MB  125 W  Mar 16th 2021        NaN
   9  Core i9-11900KF  Rocket Lake   8 / 16  3.5 to 5.3 GHz  Socket 1200      14 nm     16MB    125  Mar 16th 2021         Rs
  10   Core i9-11900T  Rocket Lake   8 / 16  1.5 to 4.9 GHz  Socket 1200      14 nm     16MB   35 W  Mar 16th 2021   Rs 35000
  11   Core i9-12900K   Alder Lake  16 / 24  3.2 to 5.2 GHz  Socket 1700      10 nm     30MB  125 W   Nov 4th 2021   Rs 58000
  12  Core i9-12900KF   Alder Lake  16 / 24  3.2 to 5.2 GHz  Socket 1700      10 nm     30MB  125 W   Nov 4th 2021   Rs 55000
```

Choice 3 - To get the latest GPUs available in the market CSVs:



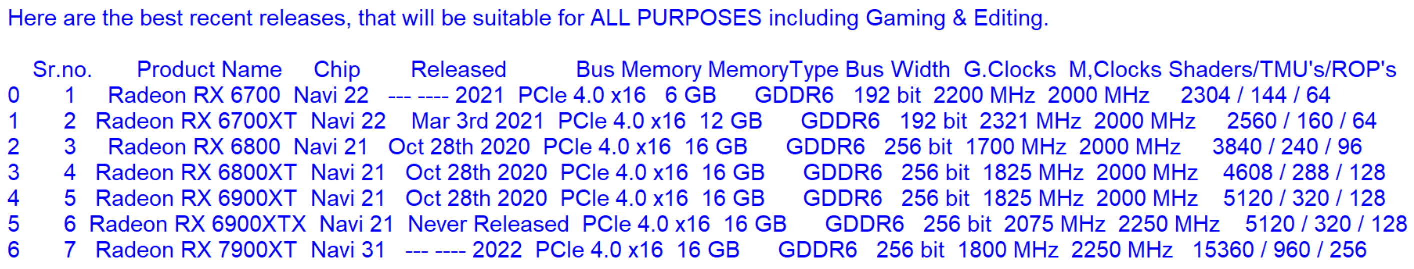
```
 elif Guide == 3:
    print("\n\nChatbot : WHICH MANUFACTURER'S RECENT GPU DO YOU WANT TO KNOW ABOUT?\n")
    GPUid=int(input("                    1. AMD\n                    2. Nvidia\n                    Enter number as reply.\n"+str(Username))
    #Storing user input in variable to compare it later in code and show predefined outputs.
```

Output:



```
Chatbot : WHICH MANUFACTURER'S RECENT GPU DO YOU WANT TO KNOW ABOUT?

                    1. AMD
                    2. Nvidia
                    Enter number as reply.
```

Choice 3 - Inner Choice 1 Output CSV:



```
Here are the best recent releases, that will be suitable for ALL PURPOSES including Gaming & Editing.

  Sr.no.      Product Name   Chip        Released          Bus Memory MemoryType Bus Width  G.Clocks  M,Clocks Shaders/TMU's/ROP's
0      1     Radeon RX 6700  Navi 22   --- ---- 2021  PCIe 4.0 x16   6 GB     GDDR6   192 bit  2200 MHz  2000 MHz     2304 / 144 / 64
1      2   Radeon RX 6700XT  Navi 22   Mar 3rd 2021  PCIe 4.0 x16  12 GB     GDDR6   192 bit  2321 MHz  2000 MHz     2560 / 160 / 64
2      3     Radeon RX 6800  Navi 21  Oct 28th 2020  PCIe 4.0 x16  16 GB     GDDR6   256 bit  1700 MHz  2000 MHz     3840 / 240 / 96
3      4   Radeon RX 6800XT  Navi 21  Oct 28th 2020  PCIe 4.0 x16  16 GB     GDDR6   256 bit  1825 MHz  2000 MHz    4608 / 288 / 128
4      5   Radeon RX 6900XT  Navi 21  Oct 28th 2020  PCIe 4.0 x16  16 GB     GDDR6   256 bit  1825 MHz  2000 MHz    5120 / 320 / 128
5      6  Radeon RX 6900XTX  Navi 21  Never Released  PCIe 4.0 x16  16 GB     GDDR6   256 bit  2075 MHz  2250 MHz    5120 / 320 / 128
6      7   Radeon RX 7900XT  Navi 31   --- ---- 2022  PCIe 4.0 x16  16 GB     GDDR6   256 bit  1800 MHz  2250 MHz   15360 / 960 / 256
```

Choice 3 - Inner Choice 2 Output CSV:

Here are the best recent releases, that will be suitable for ALL PURPOSES including Gaming & Editing.

```
   Sr.no.      Product Name   Chip      Released          Bus Memory Bus Width  G.Clocks  M,Clocks Shaders/TMU's/ROP's
0      1   GeForce RTX 3060  GA106  Jan 12th 2021  PCIe 4.0 x16  12 GB   192 bit  1320 MHz  1875 MHz     3584 / 112 / 48
1      2 GeForce RTX 3060Ti  GA104   Dec 1st 2020  PCIe 4.0 x16   8 GB   256 bit  1410 MHz  1750 MHz     4864 / 152 / 80
2      3   GeForce RTX 3070  GA104   Sep 1st 2020  PCIe 4.0 x16   8 GB   256 bit  1500 MHz  1750 MHz     5888 / 184 / 96
3      4 GeForce RTX 3070Ti  GA104  May 31st 2021  PCIe 4.0 x16   8 GB   256 bit  1575 MHz  1188 MHz     6144 / 192 / 96
4      5   GeForce RTX 3080  GA102   Sep 1st 2020  PCIe 4.0 x16  10 GB   320 bit  1440 MHz  1188 MHz     8704 / 272 / 96
5      6 GeForce RTX 3080Ti  GA102  May 31st 2021  PCIe 4.0 x16  12 GB   384 bit  1365 MHz  1188 MHz   10240 / 320 / 112
6      7   GeForce RTX 3090  GA102   Sep 1st 2020  PCIe 4.0 x16  24 GB   384 bit  1395 MHz  1219 MHz   10496 / 328 / 112
```

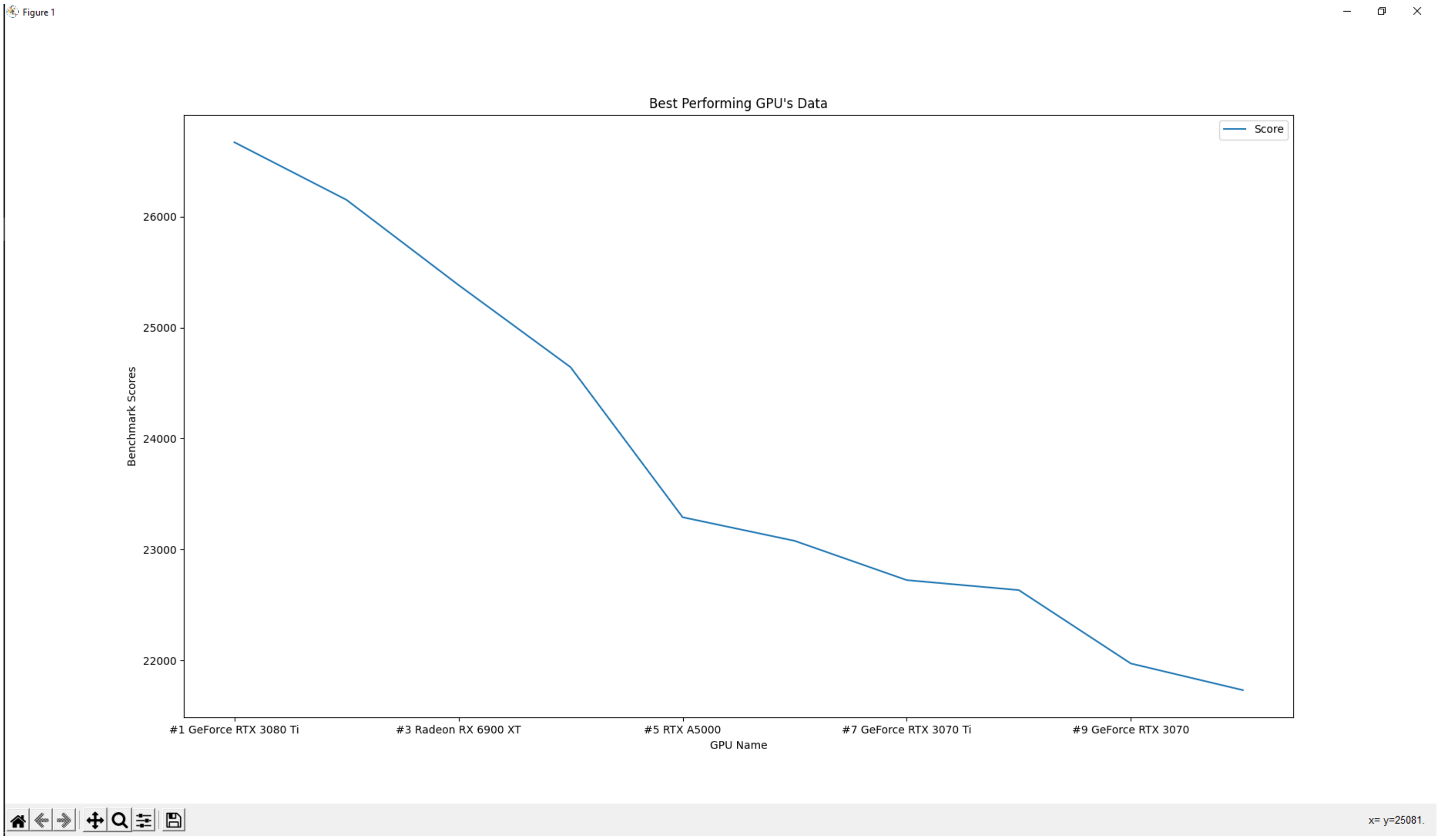
Choice 4 To Read the Visual Data of GPU CSVs:

```
elif Guide==4:
    print("\nChatbot : Get Virtual Comparison of Products sorted by name & benchmark scores.")
    TESTGPU=int(input("\n                    1. Best GPU in market\n                    2. Best CPU in market\n                    Enter number as reply
    #Storing user input in variable to compare it later in code and show predefined outputs.
```

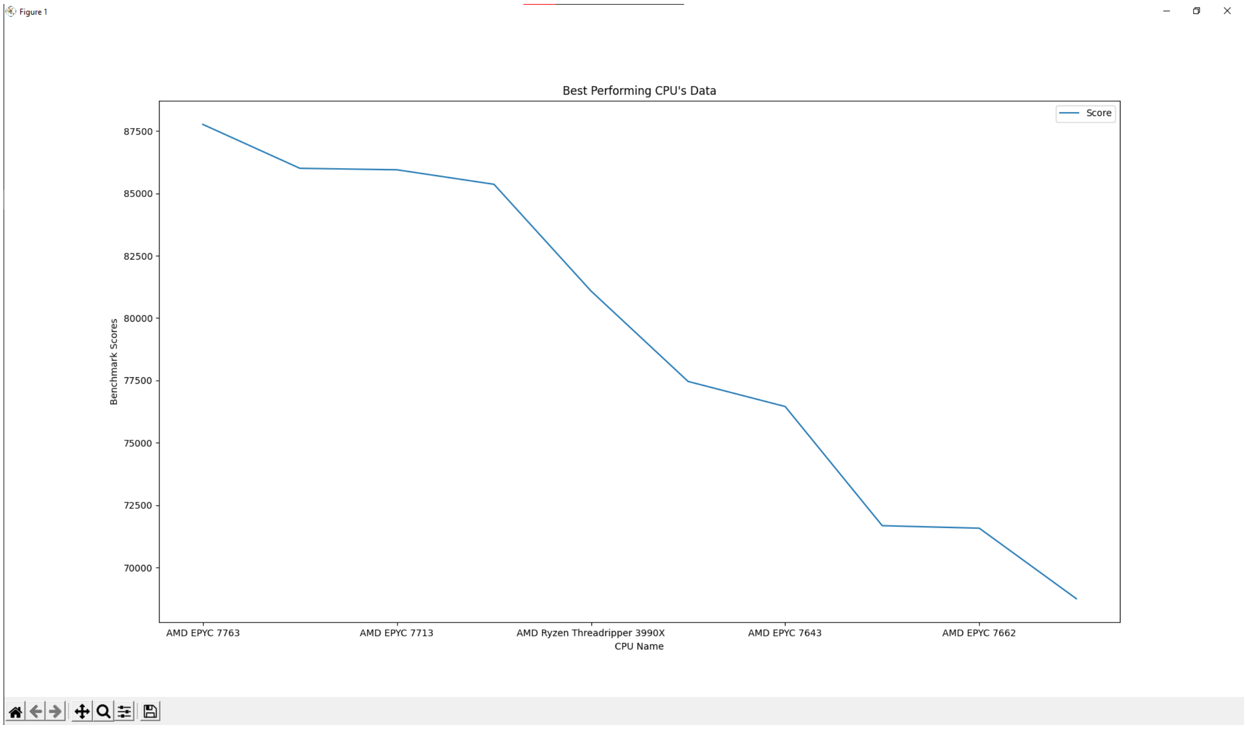
Output:

```
Chatbot : Get Virtual Comparison of Products sorted by name & benchmark scores.

                    1. Best GPU in market
                    2. Best CPU in market
                    Enter number as reply.
```

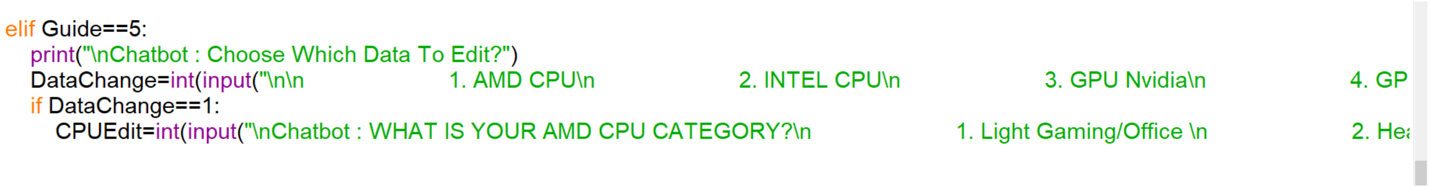
Inner Choice 1 Output of Best GPUs in the market:



Inner Choice 2 Output of Best GPUs in the market:

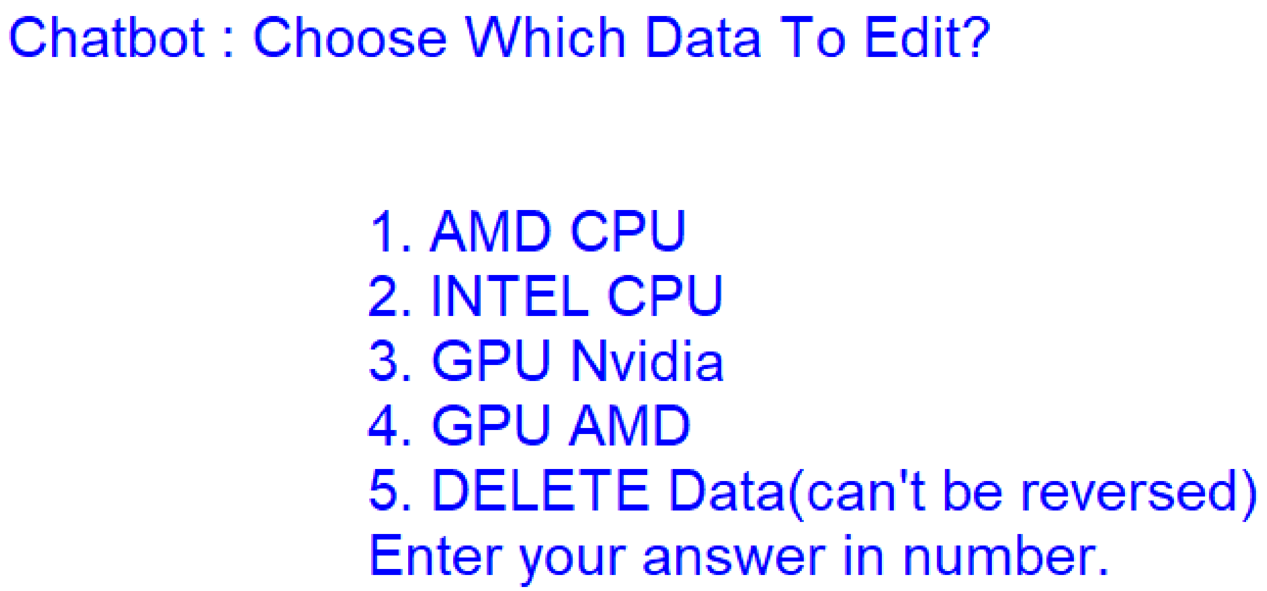


Edit Data Choice:



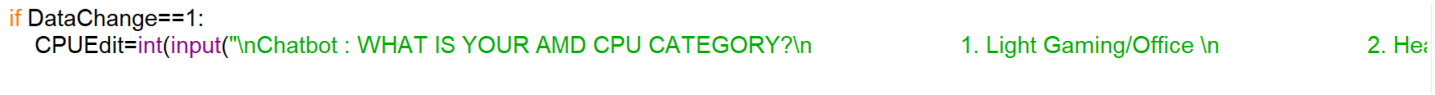
```
elif Guide==5:
    print("\nChatbot : Choose Which Data To Edit?")
    DataChange=int(input("\n\n                    1. AMD CPU\n                    2. INTEL CPU\n                    3. GPU Nvidia\n                    4. GP
    if DataChange==1:
        CPUEdit=int(input("\nChatbot : WHAT IS YOUR AMD CPU CATEGORY?\n                    1. Light Gaming/Office \n                    2. Hea
```

Output:



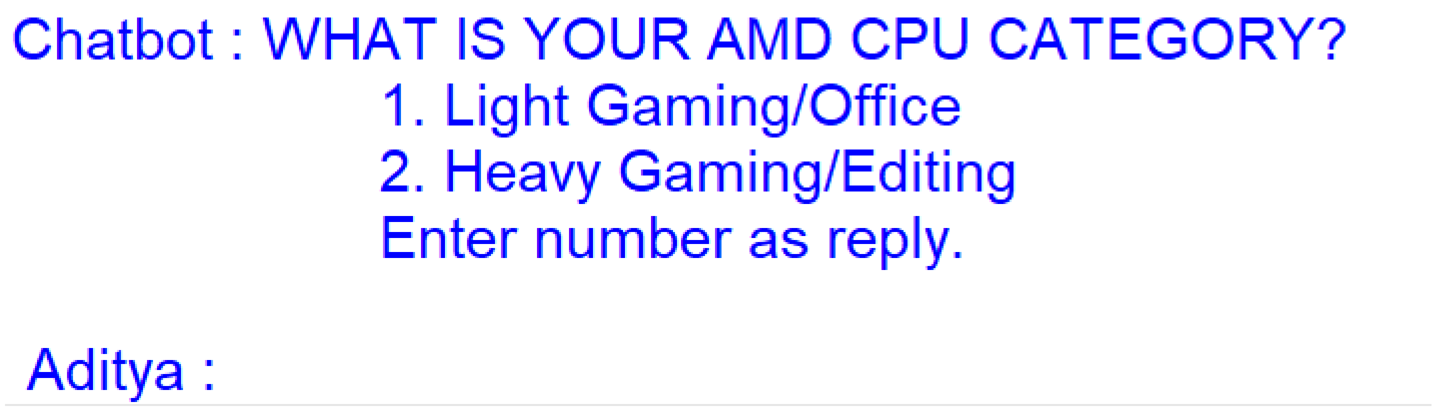
```
Chatbot : Choose Which Data To Edit?


                    1. AMD CPU
                    2. INTEL CPU
                    3. GPU Nvidia
                    4. GPU AMD
                    5. DELETE Data(can't be reversed)
                    Enter your answer in number.
```

Edit Inner Choice 1 – AMD CPU CSV edit choice:



```
if DataChange==1:
    CPUEdit=int(input("\nChatbot : WHAT IS YOUR AMD CPU CATEGORY?\n                    1. Light Gaming/Office \n                    2. Hea
```

Output:

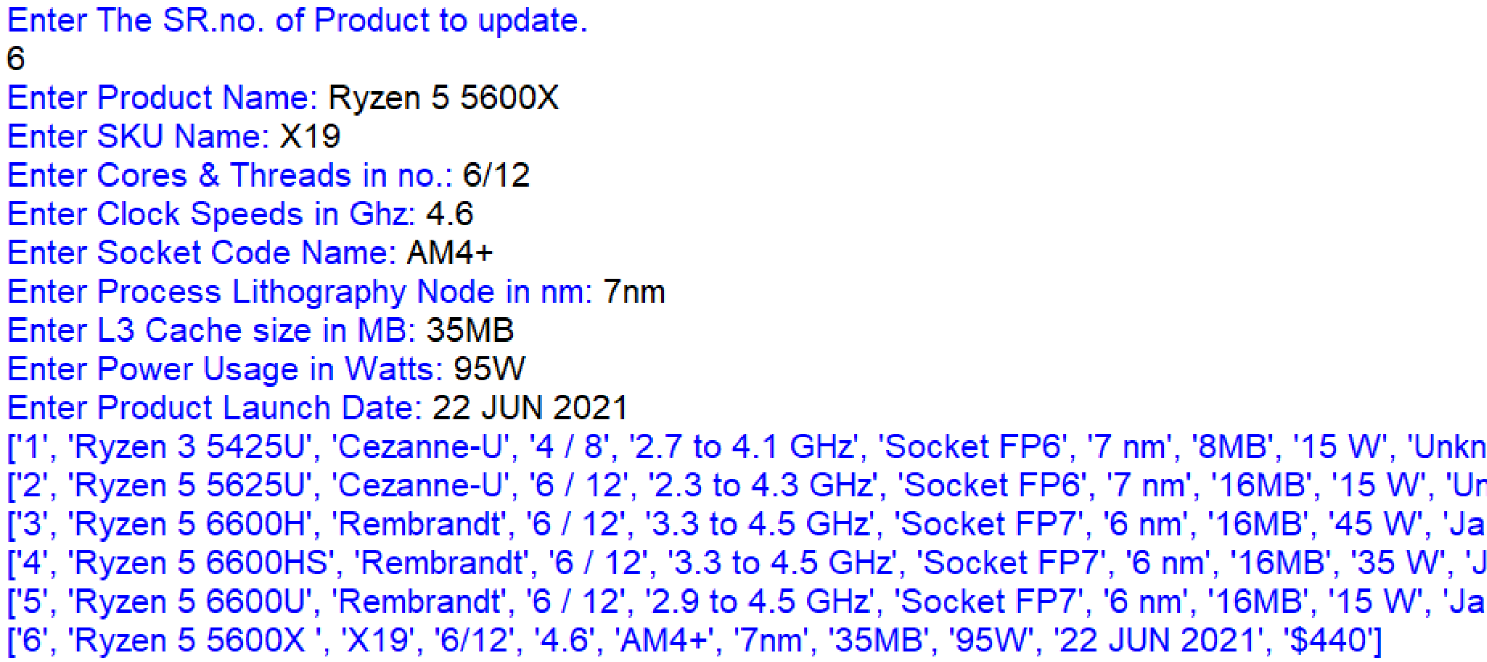


```
Chatbot : WHAT IS YOUR AMD CPU CATEGORY?
                    1. Light Gaming/Office
                    2. Heavy Gaming/Editing
                    Enter number as reply.

 Aditya :
```

Editing 1st choice of AMD CPU:

```
if CPUEdit==1:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUGAMING.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to update.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
            Product=input("Enter Product Name: ")
            row[1]=Product
            SKU=input("Enter SKU Name: ")
            row[2]=SKU
            Cores=input("Enter Cores & Threads in no.: ")
            row[3]=Cores
            Clock=input("Enter Clock Speeds in Ghz: ")
            row[4]=Clock
            Socket=input("Enter Socket Code Name: ")
            row[5]=Socket
            Lithography=input("Enter Process Lithography Node in nm: ")
            row[6]=Lithography
            Cache=input("Enter L3 Cache size in MB: ")
            row[7]=Cache
            Power=input("Enter Power Usage in Watts: ")
            row[8]=Power
            Date=input("Enter Product Launch Date: ")
            row[9]=Date
        L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUGAMING.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```

Output:

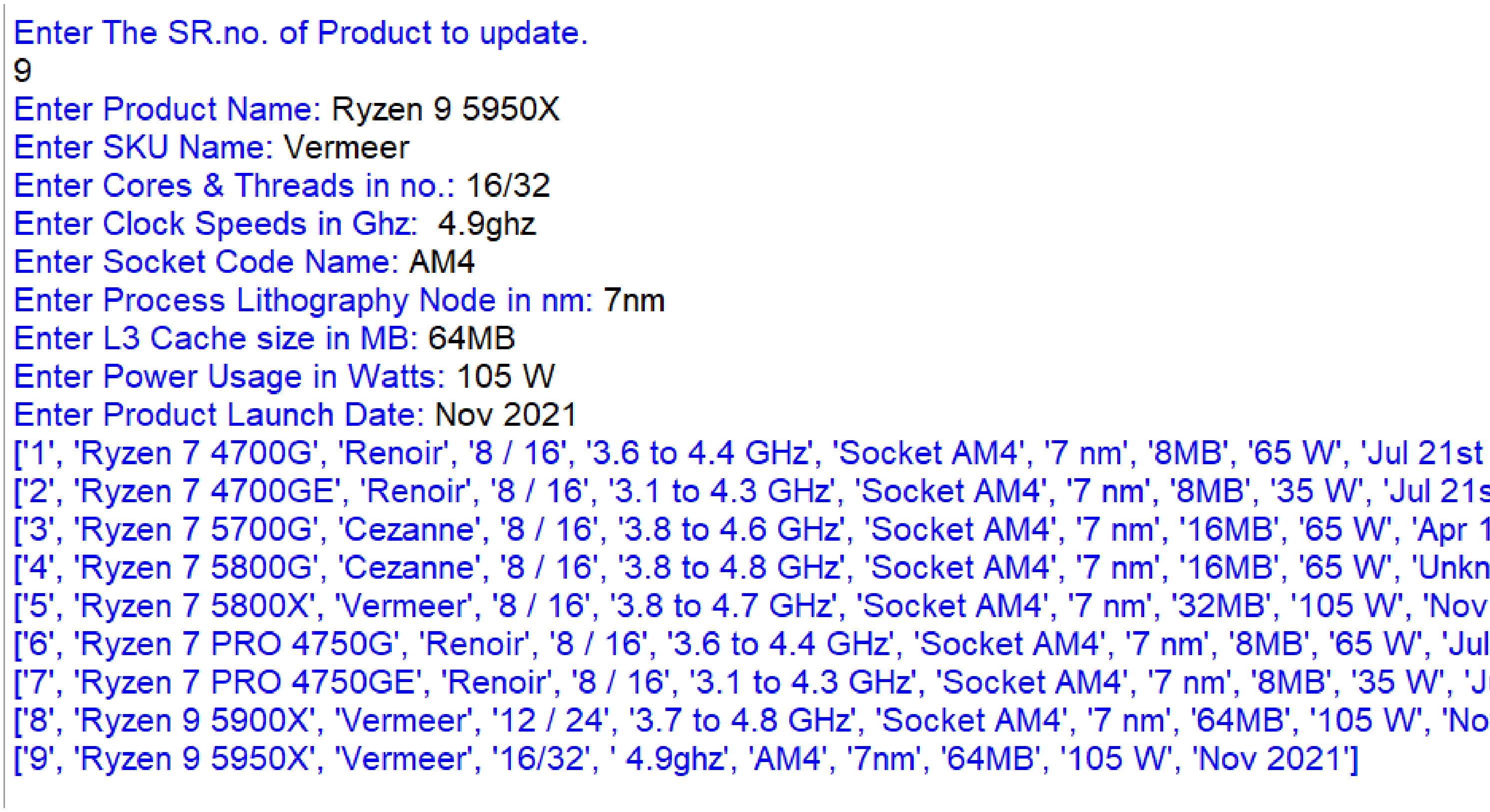


```
Enter The SR.no. of Product to update.
6
Enter Product Name: Ryzen 5 5600X
Enter SKU Name: X19
Enter Cores & Threads in no.: 6/12
Enter Clock Speeds in Ghz: 4.6
Enter Socket Code Name: AM4+
Enter Process Lithography Node in nm: 7nm
Enter L3 Cache size in MB: 35MB
Enter Power Usage in Watts: 95W
Enter Product Launch Date: 22 JUN 2021
['1', 'Ryzen 3 5425U', 'Cezanne-U', '4 / 8', '2.7 to 4.1 GHz', 'Socket FP6', '7 nm', '8MB', '15 W', 'Unkn
['2', 'Ryzen 5 5625U', 'Cezanne-U', '6 / 12', '2.3 to 4.3 GHz', 'Socket FP6', '7 nm', '16MB', '15 W', 'Ur
['3', 'Ryzen 5 6600H', 'Rembrandt', '6 / 12', '3.3 to 4.5 GHz', 'Socket FP7', '6 nm', '16MB', '45 W', 'Ja
['4', 'Ryzen 5 6600HS', 'Rembrandt', '6 / 12', '3.3 to 4.5 GHz', 'Socket FP7', '6 nm', '16MB', '35 W', 'J
['5', 'Ryzen 5 6600U', 'Rembrandt', '6 / 12', '2.9 to 4.5 GHz', 'Socket FP7', '6 nm', '16MB', '15 W', 'Ja
['6', 'Ryzen 5 5600X ', 'X19', '6/12', '4.6', 'AM4+', '7nm', '35MB', '95W', '22 JUN 2021', '$440']
```

Editing 2ND choice of AMD CPU:

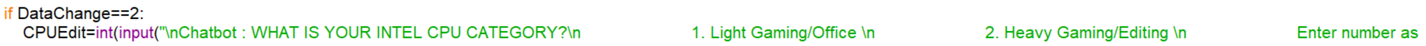
```
if CPUEdit==2:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUEDITING.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to update.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
            Product=input("Enter Product Name: ")
            row[1]=Product
            SKU=input("Enter SKU Name: ")
            row[2]=SKU
            Cores=input("Enter Cores & Threads in no.: ")
            row[3]=Cores
            Clock=input("Enter Clock Speeds in Ghz: ")
            row[4]=Clock
            Socket=input("Enter Socket Code Name: ")
            row[5]=Socket
            Lithography=input("Enter Process Lithography Node in nm: ")
            row[6]=Lithography
            Cache=input("Enter L3 Cache size in MB: ")
            row[7]=Cache
            Power=input("Enter Power Usage in Watts: ")
            row[8]=Power
            Date=input("Enter Product Launch Date: ")
            row[9]=Date
        L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUEDITING.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```

Output:



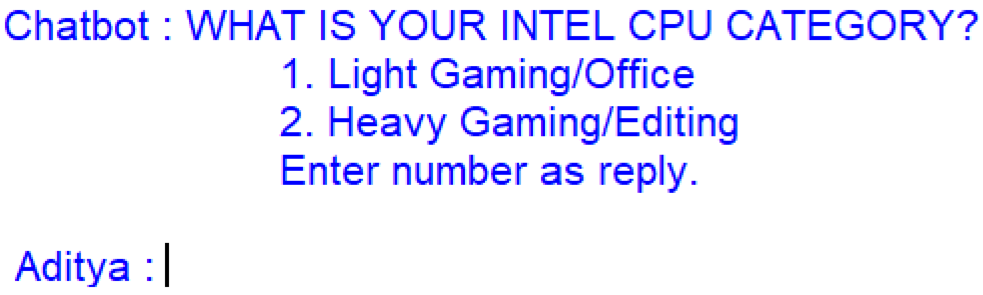
```
Enter The SR.no. of Product to update.
9
Enter Product Name: Ryzen 9 5950X
Enter SKU Name: Vermeer
Enter Cores & Threads in no.: 16/32
Enter Clock Speeds in Ghz:  4.9ghz
Enter Socket Code Name: AM4
Enter Process Lithography Node in nm: 7nm
Enter L3 Cache size in MB: 64MB
Enter Power Usage in Watts: 105 W
Enter Product Launch Date: Nov 2021
['1', 'Ryzen 7 4700G', 'Renoir', '8 / 16', '3.6 to 4.4 GHz', 'Socket AM4', '7 nm', '8MB', '65 W', 'Jul 21st
['2', 'Ryzen 7 4700GE', 'Renoir', '8 / 16', '3.1 to 4.3 GHz', 'Socket AM4', '7 nm', '8MB', '35 W', 'Jul 21s
['3', 'Ryzen 7 5700G', 'Cezanne', '8 / 16', '3.8 to 4.6 GHz', 'Socket AM4', '7 nm', '16MB', '65 W', 'Apr 1
['4', 'Ryzen 7 5800G', 'Cezanne', '8 / 16', '3.8 to 4.8 GHz', 'Socket AM4', '7 nm', '16MB', '65 W', 'Unkn
['5', 'Ryzen 7 5800X', 'Vermeer', '8 / 16', '3.8 to 4.7 GHz', 'Socket AM4', '7 nm', '32MB', '105 W', 'Nov
['6', 'Ryzen 7 PRO 4750G', 'Renoir', '8 / 16', '3.6 to 4.4 GHz', 'Socket AM4', '7 nm', '8MB', '65 W', 'Jul
['7', 'Ryzen 7 PRO 4750GE', 'Renoir', '8 / 16', '3.1 to 4.3 GHz', 'Socket AM4', '7 nm', '8MB', '35 W', 'J
['8', 'Ryzen 9 5900X', 'Vermeer', '12 / 24', '3.7 to 4.8 GHz', 'Socket AM4', '7 nm', '64MB', '105 W', 'No
['9', 'Ryzen 9 5950X', 'Vermeer', '16/32', ' 4.9ghz', 'AM4', '7nm', '64MB', '105 W', 'Nov 2021']
```

Edit Inner Choice 2 – INTEL CPU CSV edit choice:



```
if DataChange==2:
    CPUEdit=int(input("\nChatbot : WHAT IS YOUR INTEL CPU CATEGORY?\n                1. Light Gaming/Office \n                2. Heavy Gaming/Editing \n                Enter number as
```

Output:

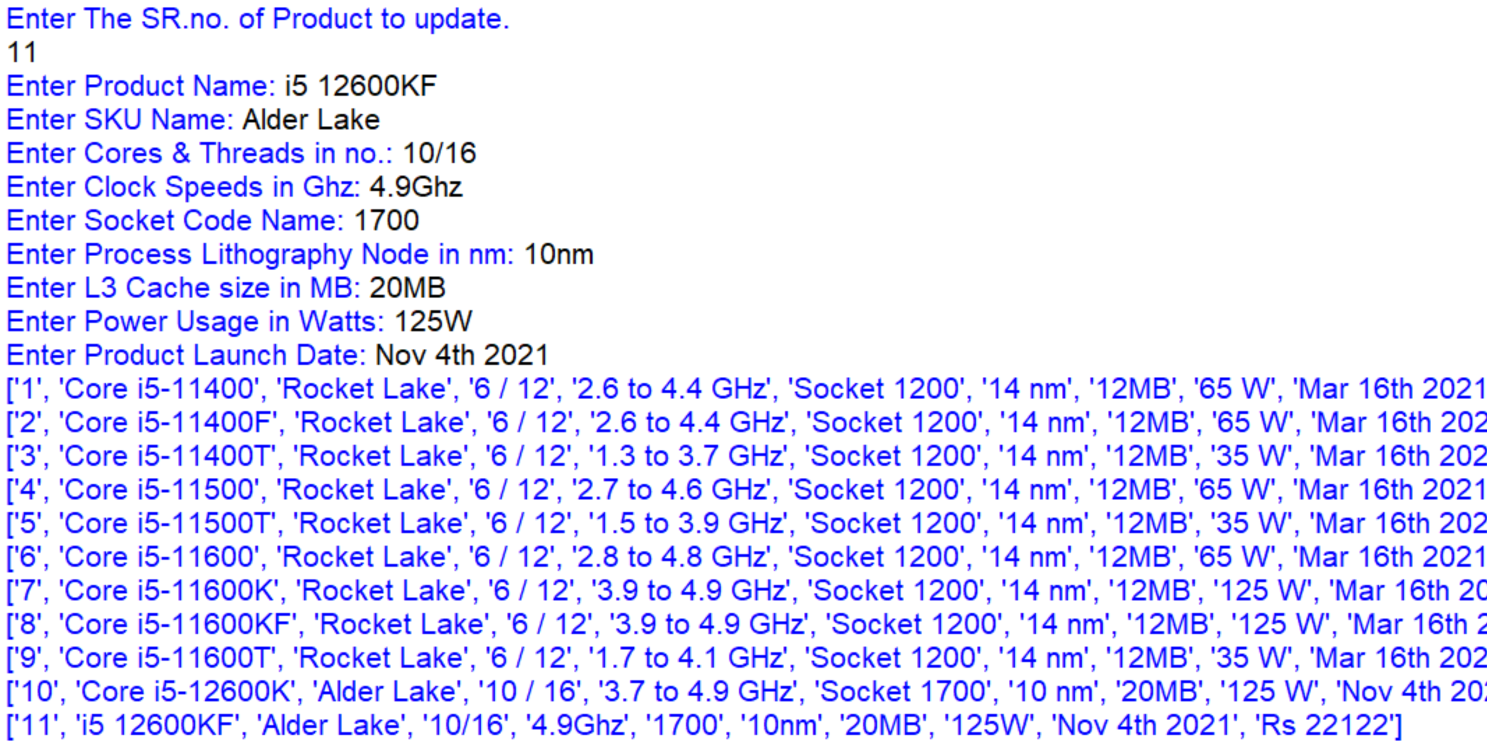


```
Chatbot : WHAT IS YOUR INTEL CPU CATEGORY?
                1. Light Gaming/Office
                2. Heavy Gaming/Editing
                Enter number as reply.

 Aditya :|
```

Editing 1st choice of Intel CPU:

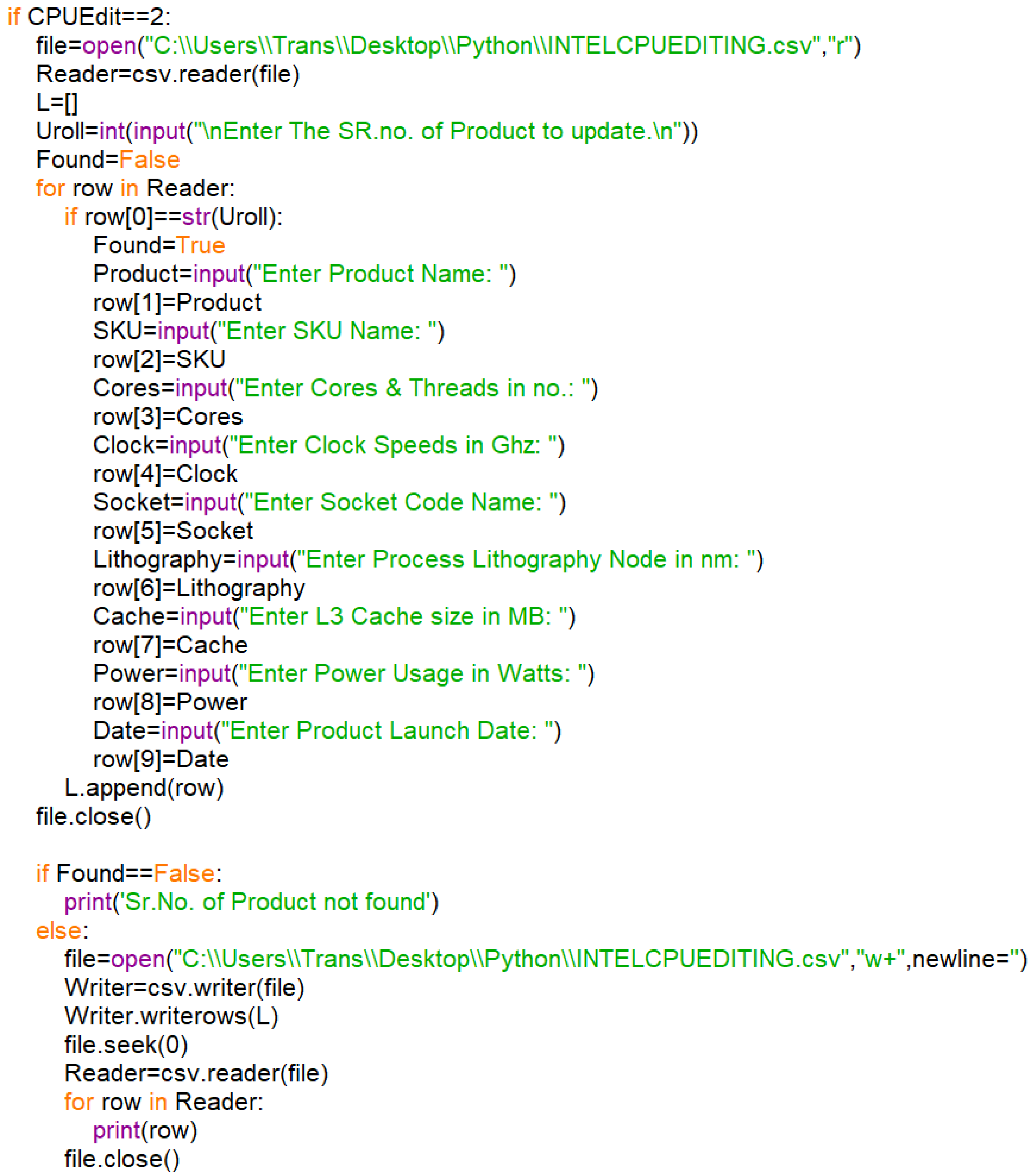
```
if CPUEdit==1:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUGAMING.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to update.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
            Product=input("Enter Product Name: ")
            row[1]=Product
            SKU=input("Enter SKU Name: ")
            row[2]=SKU
            Cores=input("Enter Cores & Threads in no.: ")
            row[3]=Cores
            Clock=input("Enter Clock Speeds in Ghz: ")
            row[4]=Clock
            Socket=input("Enter Socket Code Name: ")
            row[5]=Socket
            Lithography=input("Enter Process Lithography Node in nm: ")
            row[6]=Lithography
            Cache=input("Enter L3 Cache size in MB: ")
            row[7]=Cache
            Power=input("Enter Power Usage in Watts: ")
            row[8]=Power
            Date=input("Enter Product Launch Date: ")
            row[9]=Date
        L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUGAMING.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```

Output:



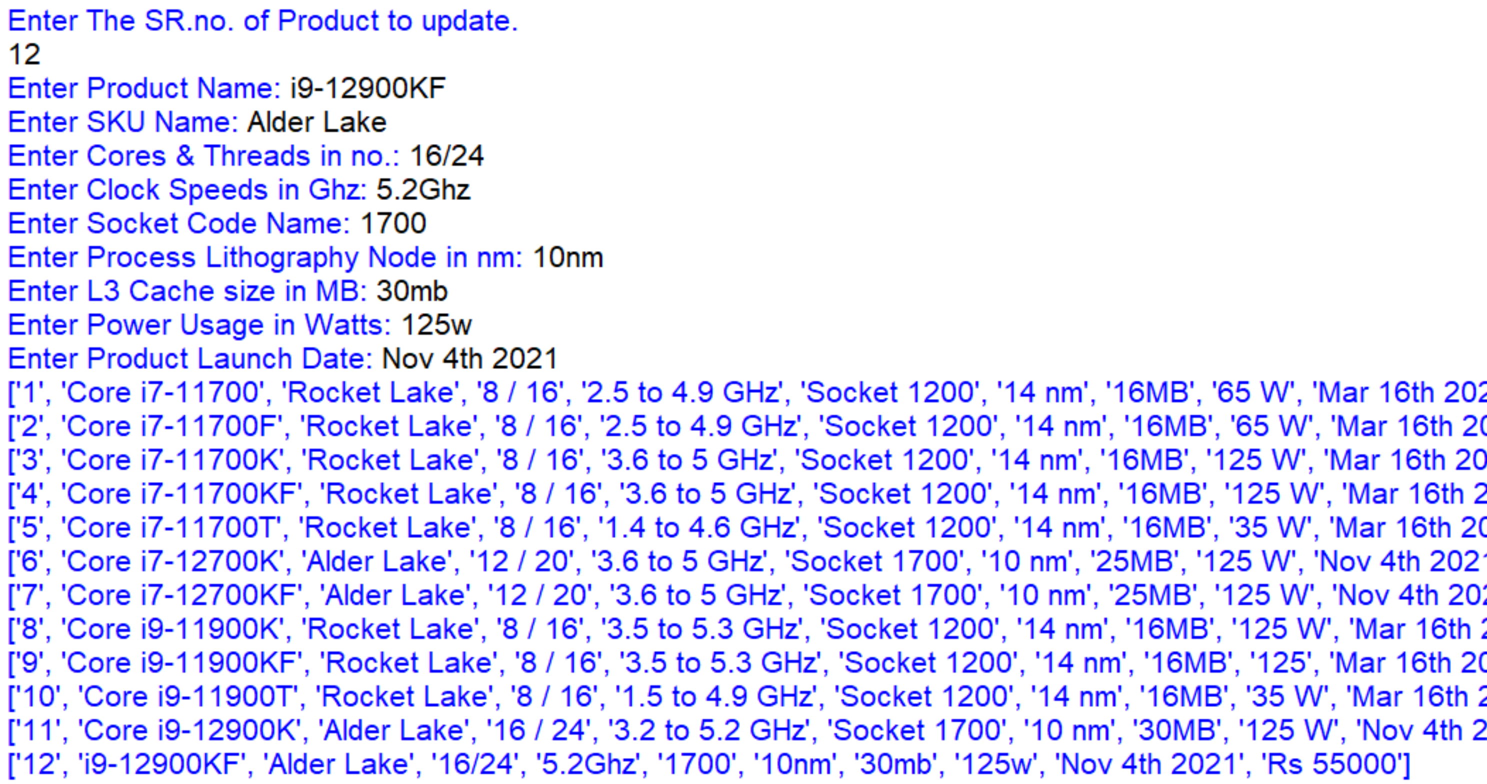
```
Enter The SR.no. of Product to update.
11
Enter Product Name: i5 12600KF
Enter SKU Name: Alder Lake
Enter Cores & Threads in no.: 10/16
Enter Clock Speeds in Ghz: 4.9Ghz
Enter Socket Code Name: 1700
Enter Process Lithography Node in nm: 10nm
Enter L3 Cache size in MB: 20MB
Enter Power Usage in Watts: 125W
Enter Product Launch Date: Nov 4th 2021
['1', 'Core i5-11400', 'Rocket Lake', '6 / 12', '2.6 to 4.4 GHz', 'Socket 1200', '14 nm', '12MB', '65 W', 'Mar 16th 2021
['2', 'Core i5-11400F', 'Rocket Lake', '6 / 12', '2.6 to 4.4 GHz', 'Socket 1200', '14 nm', '12MB', '65 W', 'Mar 16th 202
['3', 'Core i5-11400T', 'Rocket Lake', '6 / 12', '1.3 to 3.7 GHz', 'Socket 1200', '14 nm', '12MB', '35 W', 'Mar 16th 202
['4', 'Core i5-11500', 'Rocket Lake', '6 / 12', '2.7 to 4.6 GHz', 'Socket 1200', '14 nm', '12MB', '65 W', 'Mar 16th 2021
['5', 'Core i5-11500T', 'Rocket Lake', '6 / 12', '1.5 to 3.9 GHz', 'Socket 1200', '14 nm', '12MB', '35 W', 'Mar 16th 202
['6', 'Core i5-11600', 'Rocket Lake', '6 / 12', '2.8 to 4.8 GHz', 'Socket 1200', '14 nm', '12MB', '65 W', 'Mar 16th 2021
['7', 'Core i5-11600K', 'Rocket Lake', '6 / 12', '3.9 to 4.9 GHz', 'Socket 1200', '14 nm', '12MB', '125 W', 'Mar 16th 20
['8', 'Core i5-11600KF', 'Rocket Lake', '6 / 12', '3.9 to 4.9 GHz', 'Socket 1200', '14 nm', '12MB', '125 W', 'Mar 16th 2
['9', 'Core i5-11600T', 'Rocket Lake', '6 / 12', '1.7 to 4.1 GHz', 'Socket 1200', '14 nm', '12MB', '35 W', 'Mar 16th 202
['10', 'Core i5-12600K', 'Alder Lake', '10 / 16', '3.7 to 4.9 GHz', 'Socket 1700', '10 nm', '20MB', '125 W', 'Nov 4th 202
['11', 'i5 12600KF', 'Alder Lake', '10/16', '4.9Ghz', '1700', '10nm', '20MB', '125W', 'Nov 4th 2021', 'Rs 22122']
```

Editing 2nd choice of Intel CPU:



```
if CPUEdit==2:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUEDITING.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to update.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
            Product=input("Enter Product Name: ")
            row[1]=Product
            SKU=input("Enter SKU Name: ")
            row[2]=SKU
            Cores=input("Enter Cores & Threads in no.: ")
            row[3]=Cores
            Clock=input("Enter Clock Speeds in Ghz: ")
            row[4]=Clock
            Socket=input("Enter Socket Code Name: ")
            row[5]=Socket
            Lithography=input("Enter Process Lithography Node in nm: ")
            row[6]=Lithography
            Cache=input("Enter L3 Cache size in MB: ")
            row[7]=Cache
            Power=input("Enter Power Usage in Watts: ")
            row[8]=Power
            Date=input("Enter Product Launch Date: ")
            row[9]=Date
        L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUEDITING.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```

Output:

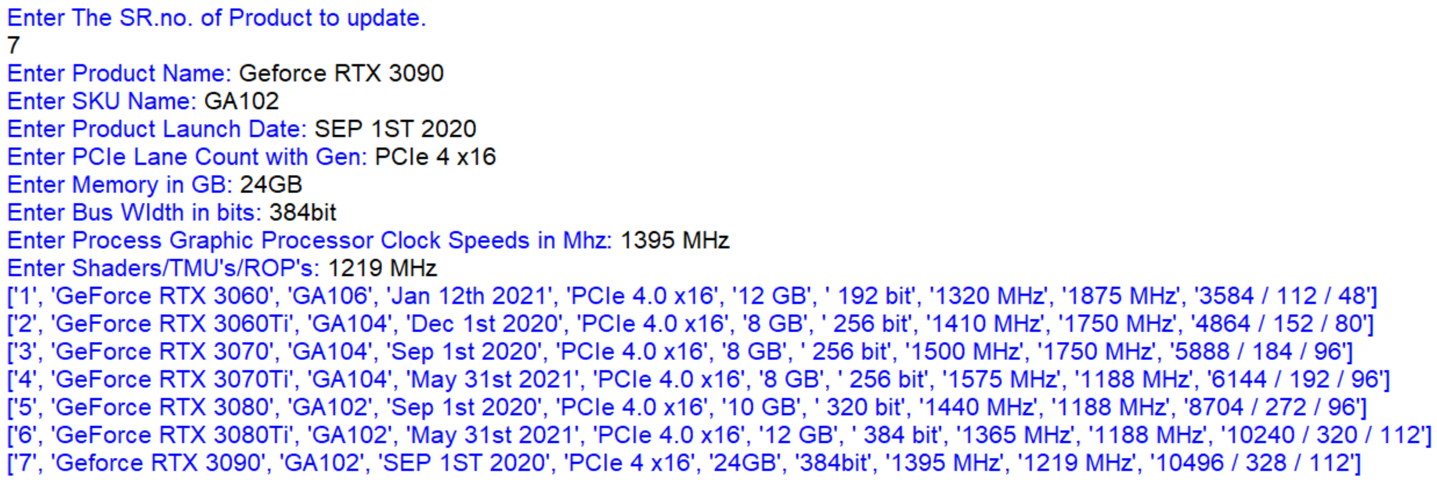


```
Enter The SR.no. of Product to update.
12
Enter Product Name: i9-12900KF
Enter SKU Name: Alder Lake
Enter Cores & Threads in no.: 16/24
Enter Clock Speeds in Ghz: 5.2Ghz
Enter Socket Code Name: 1700
Enter Process Lithography Node in nm: 10nm
Enter L3 Cache size in MB: 30mb
Enter Power Usage in Watts: 125w
Enter Product Launch Date: Nov 4th 2021
['1', 'Core i7-11700', 'Rocket Lake', '8 / 16', '2.5 to 4.9 GHz', 'Socket 1200', '14 nm', '16MB', '65 W', 'Mar 16th 202
['2', 'Core i7-11700F', 'Rocket Lake', '8 / 16', '2.5 to 4.9 GHz', 'Socket 1200', '14 nm', '16MB', '65 W', 'Mar 16th 20
['3', 'Core i7-11700K', 'Rocket Lake', '8 / 16', '3.6 to 5 GHz', 'Socket 1200', '14 nm', '16MB', '125 W', 'Mar 16th 20
['4', 'Core i7-11700KF', 'Rocket Lake', '8 / 16', '3.6 to 5 GHz', 'Socket 1200', '14 nm', '16MB', '125 W', 'Mar 16th 2
['5', 'Core i7-11700T', 'Rocket Lake', '8 / 16', '1.4 to 4.6 GHz', 'Socket 1200', '14 nm', '16MB', '35 W', 'Mar 16th 20
['6', 'Core i7-12700K', 'Alder Lake', '12 / 20', '3.6 to 5 GHz', 'Socket 1700', '10 nm', '25MB', '125 W', 'Nov 4th 2021
['7', 'Core i7-12700KF', 'Alder Lake', '12 / 20', '3.6 to 5 GHz', 'Socket 1700', '10 nm', '25MB', '125 W', 'Nov 4th 202
['8', 'Core i9-11900K', 'Rocket Lake', '8 / 16', '3.5 to 5.3 GHz', 'Socket 1200', '14 nm', '16MB', '125 W', 'Mar 16th 2
['9', 'Core i9-11900KF', 'Rocket Lake', '8 / 16', '3.5 to 5.3 GHz', 'Socket 1200', '14 nm', '16MB', '125', 'Mar 16th 20
['10', 'Core i9-11900T', 'Rocket Lake', '8 / 16', '1.5 to 4.9 GHz', 'Socket 1200', '14 nm', '16MB', '35 W', 'Mar 16th 2
['11', 'Core i9-12900K', 'Alder Lake', '16 / 24', '3.2 to 5.2 GHz', 'Socket 1700', '10 nm', '30MB', '125 W', 'Nov 4th 2
['12', 'i9-12900KF', 'Alder Lake', '16/24', '5.2Ghz', '1700', '10nm', '30mb', '125w', 'Nov 4th 2021', 'Rs 55000']
```

Edit Inner Choice 3 - NVIDIA GPUs CSV edit choice:

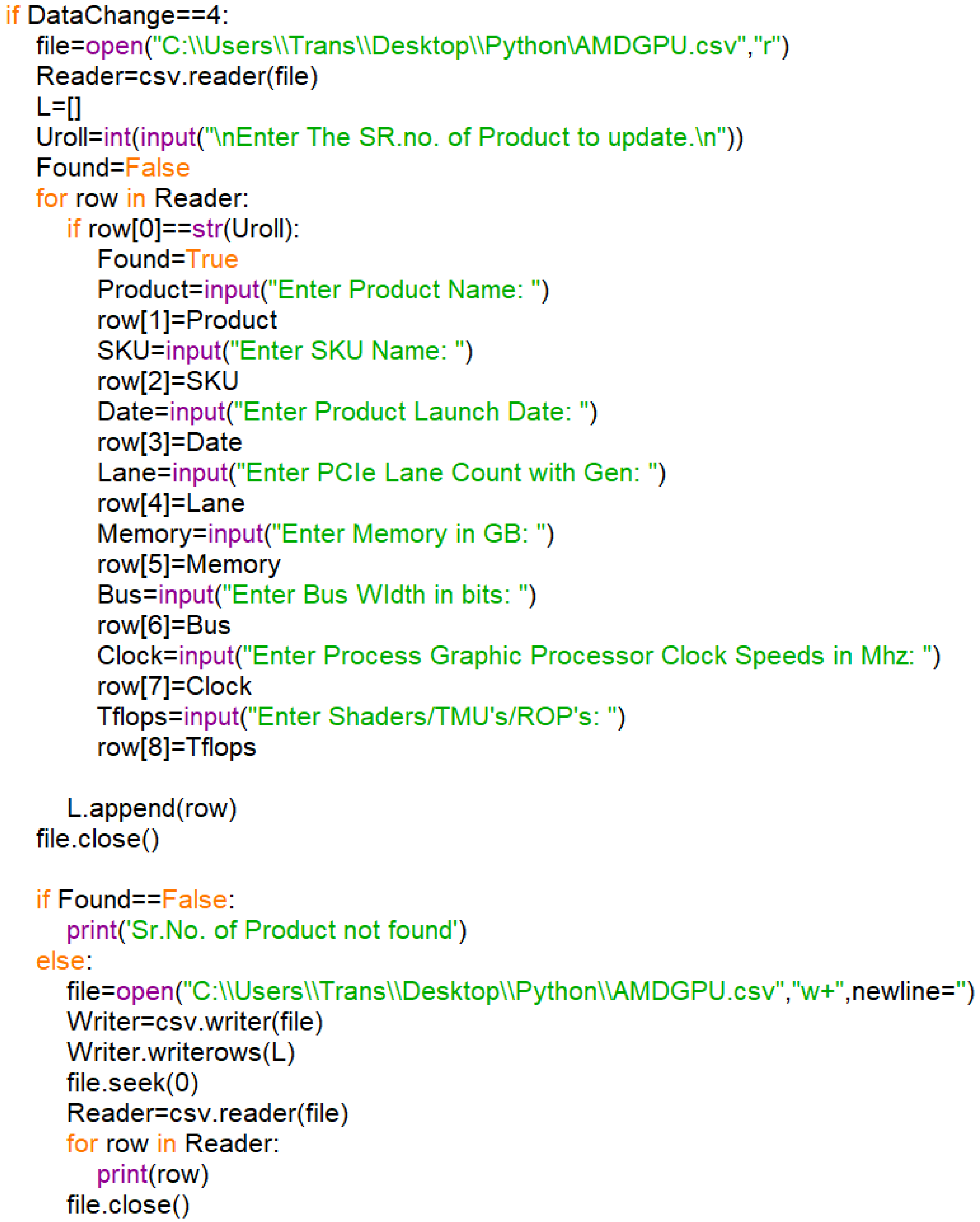
```
if DataChange==3:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\NVIDIAGPU.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to update.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
            Product=input("Enter Product Name: ")
            row[1]=Product
            SKU=input("Enter SKU Name: ")
            row[2]=SKU
            Date=input("Enter Product Launch Date: ")
            row[3]=Date
            Lane=input("Enter PCIe Lane Count with Gen: ")
            row[4]=Lane
            Memory=input("Enter Memory in GB: ")
            row[5]=Memory
            Bus=input("Enter Bus WIdth in bits: ")
            row[6]=Bus
            Clock=input("Enter Process Graphic Processor Clock Speeds in Mhz: ")
            row[7]=Clock
            Tflops=input("Enter Shaders/TMU's/ROP's: ")
            row[8]=Tflops

        L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\NVIDIAGPU.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```

Output:



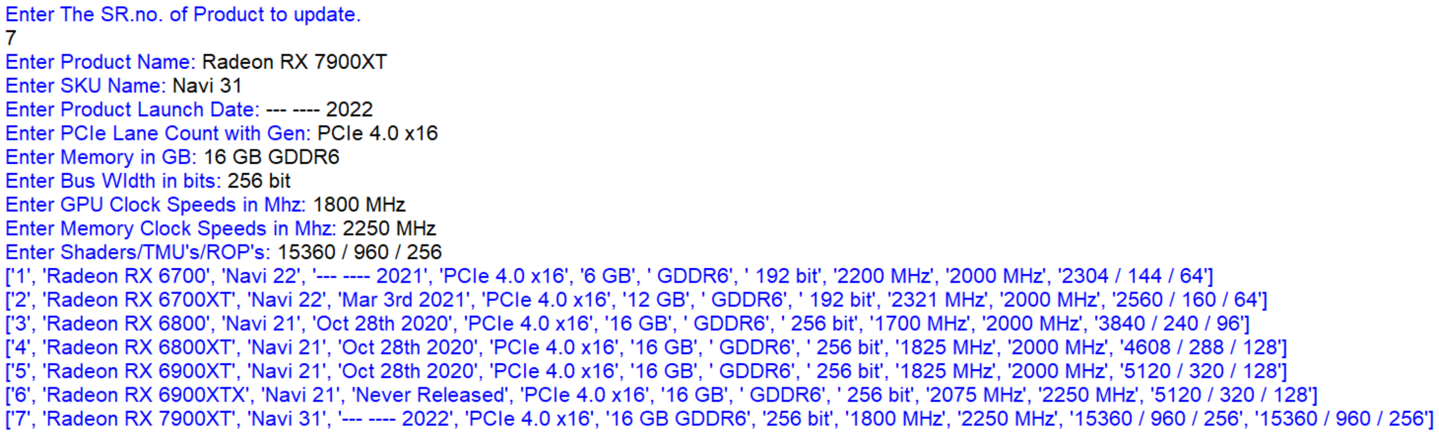
```
Enter The SR.no. of Product to update.
7
Enter Product Name: Geforce RTX 3090
Enter SKU Name: GA102
Enter Product Launch Date: SEP 1ST 2020
Enter PCIe Lane Count with Gen: PCIe 4 x16
Enter Memory in GB: 24GB
Enter Bus WIdth in bits: 384bit
Enter Process Graphic Processor Clock Speeds in Mhz: 1395 MHz
Enter Shaders/TMU's/ROP's: 1219 MHz
['1', 'GeForce RTX 3060', 'GA106', 'Jan 12th 2021', 'PCIe 4.0 x16', '12 GB', ' 192 bit', '1320 MHz', '1875 MHz', '3584 / 112 / 48']
['2', 'GeForce RTX 3060Ti', 'GA104', 'Dec 1st 2020', 'PCIe 4.0 x16', '8 GB', ' 256 bit', '1410 MHz', '1750 MHz', '4864 / 152 / 80']
['3', 'GeForce RTX 3070', 'GA104', 'Sep 1st 2020', 'PCIe 4.0 x16', '8 GB', ' 256 bit', '1500 MHz', '1750 MHz', '5888 / 184 / 96']
['4', 'GeForce RTX 3070Ti', 'GA104', 'May 31st 2021', 'PCIe 4.0 x16', '8 GB', ' 256 bit', '1575 MHz', '1188 MHz', '6144 / 192 / 96']
['5', 'GeForce RTX 3080', 'GA102', 'Sep 1st 2020', 'PCIe 4.0 x16', '10 GB', ' 320 bit', '1440 MHz', '1188 MHz', '8704 / 272 / 96']
['6', 'GeForce RTX 3080Ti', 'GA102', 'May 31st 2021', 'PCIe 4.0 x16', '12 GB', ' 384 bit', '1365 MHz', '1188 MHz', '10240 / 320 / 112']
['7', 'Geforce RTX 3090', 'GA102', 'SEP 1ST 2020', 'PCIe 4 x16', '24GB', '384bit', '1395 MHz', '1219 MHz', '10496 / 328 / 112']
```

Edit Inner Choice 4 - AMD GPUs CSV edit choice:



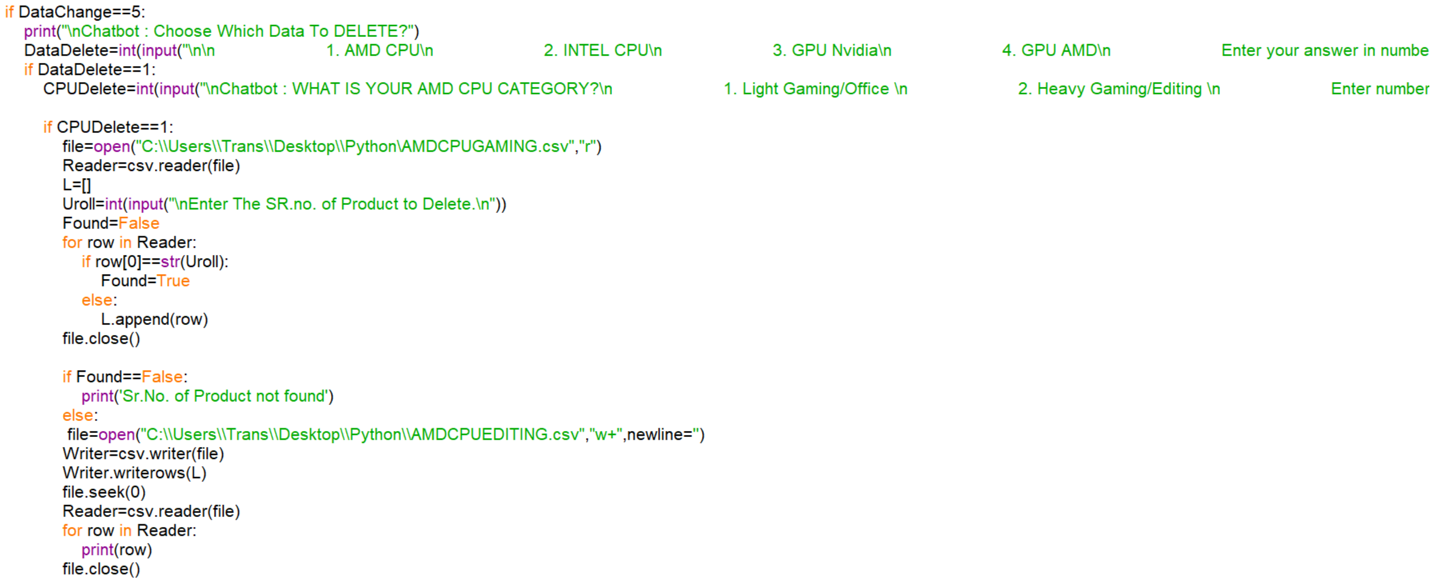
```
if DataChange==4:
    file=open("C:\\Users\\Trans\\Desktop\\Python\AMDGPU.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to update.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
            Product=input("Enter Product Name: ")
            row[1]=Product
            SKU=input("Enter SKU Name: ")
            row[2]=SKU
            Date=input("Enter Product Launch Date: ")
            row[3]=Date
            Lane=input("Enter PCIe Lane Count with Gen: ")
            row[4]=Lane
            Memory=input("Enter Memory in GB: ")
            row[5]=Memory
            Bus=input("Enter Bus WIdth in bits: ")
            row[6]=Bus
            Clock=input("Enter Process Graphic Processor Clock Speeds in Mhz: ")
            row[7]=Clock
            Tflops=input("Enter Shaders/TMU's/ROP's: ")
            row[8]=Tflops

        L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDGPU.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```

Output:

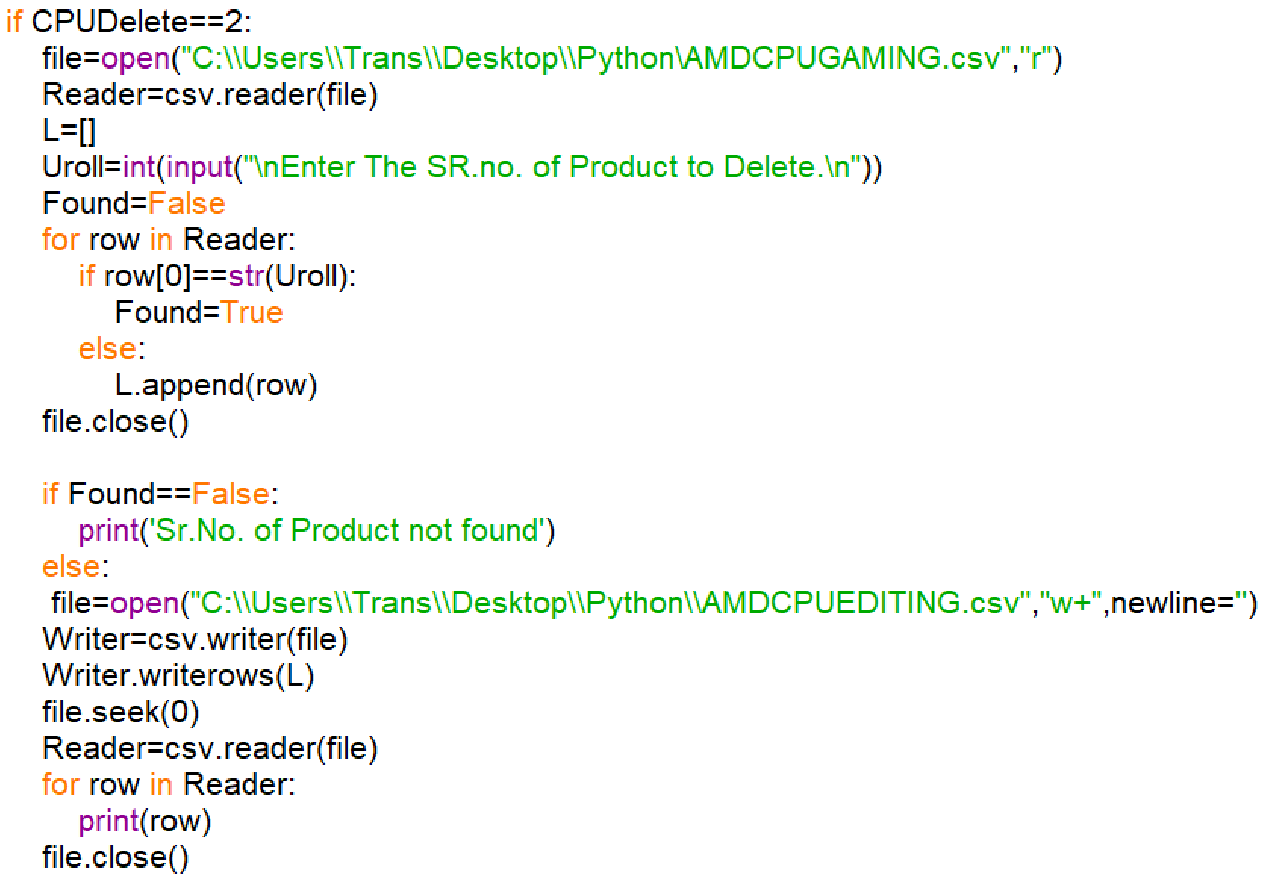


```
Enter The SR.no. of Product to update.
7
Enter Product Name: Radeon RX 7900XT
Enter SKU Name: Navi 31
Enter Product Launch Date: --- ---- 2022
Enter PCIe Lane Count with Gen: PCIe 4.0 x16
Enter Memory in GB: 16 GB GDDR6
Enter Bus WIdth in bits: 256 bit
Enter GPU Clock Speeds in Mhz: 1800 MHz
Enter Memory Clock Speeds in Mhz: 2250 MHz
Enter Shaders/TMU's/ROP's: 15360 / 960 / 256
['1', 'Radeon RX 6700', 'Navi 22', '--- ---- 2021', 'PCIe 4.0 x16', '6 GB', ' GDDR6', ' 192 bit', '2200 MHz', '2000 MHz', '2304 / 144 / 64']
['2', 'Radeon RX 6700XT', 'Navi 22', 'Mar 3rd 2021', 'PCIe 4.0 x16', '12 GB', ' GDDR6', ' 192 bit', '2321 MHz', '2000 MHz', '2560 / 160 / 64']
['3', 'Radeon RX 6800', 'Navi 21', 'Oct 28th 2020', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '1700 MHz', '2000 MHz', '3840 / 240 / 96']
['4', 'Radeon RX 6800XT', 'Navi 21', 'Oct 28th 2020', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '1825 MHz', '2000 MHz', '4608 / 288 / 128']
['5', 'Radeon RX 6900XT', 'Navi 21', 'Oct 28th 2020', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '1825 MHz', '2000 MHz', '5120 / 320 / 128']
['6', 'Radeon RX 6900XTX', 'Navi 21', 'Never Released', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '2075 MHz', '2250 MHz', '5120 / 320 / 128']
['7', 'Radeon RX 7900XT', 'Navi 31', '--- ---- 2022', 'PCIe 4.0 x16', '16 GB GDDR6', '256 bit', '1800 MHz', '2250 MHz', '15360 / 960 / 256', '15360 / 960 / 256']
```

Edit Inner Choice 5 - Delete any data choice:

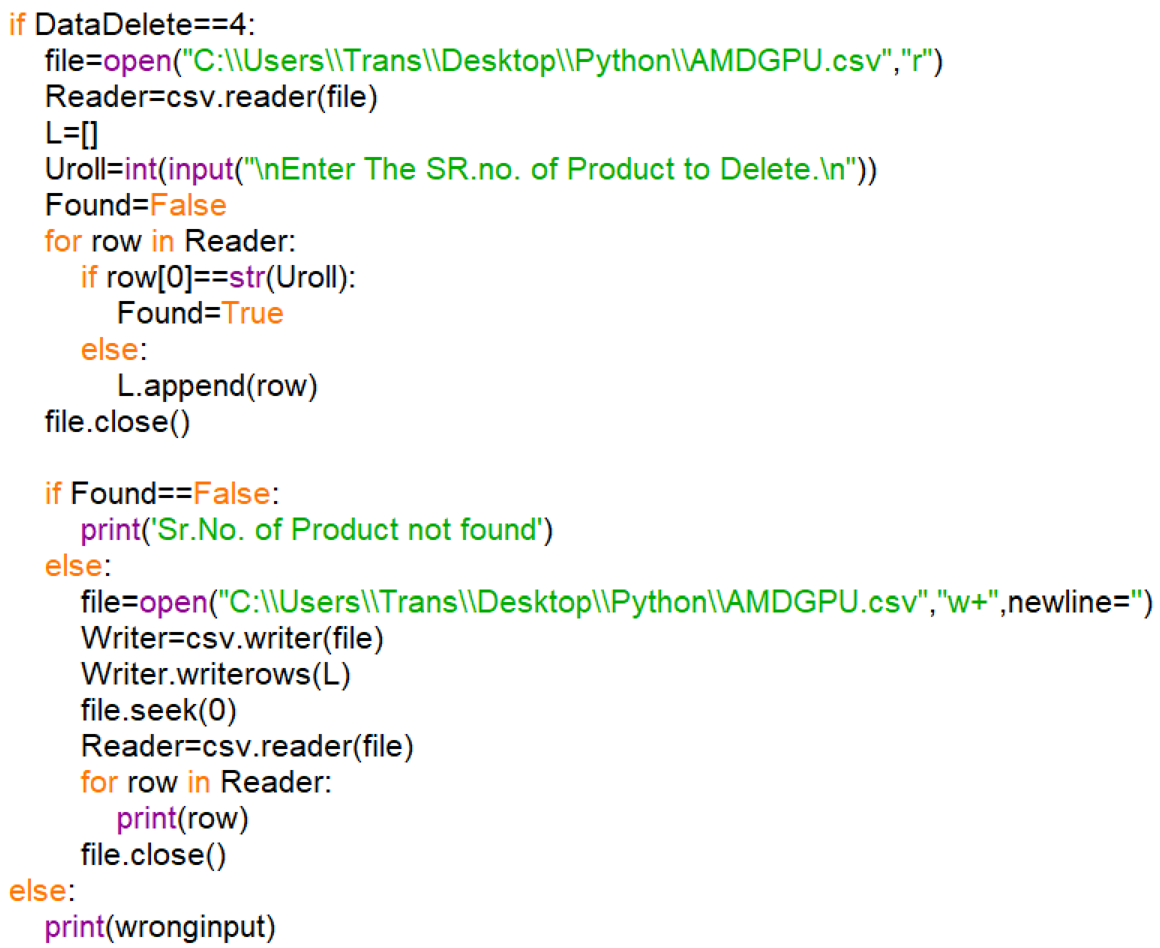


```
if DataChange==5:
    print("\nChatbot : Choose Which Data To DELETE?")
    DataDelete=int(input("\n\n                    1. AMD CPU\n                    2. INTEL CPU\n                    3. GPU Nvidia\n                    4. GPU AMD\n                    Enter your answer in numbe
    if DataDelete==1:
        CPUDelete=int(input("\nChatbot : WHAT IS YOUR AMD CPU CATEGORY?\n                    1. Light Gaming/Office \n                    2. Heavy Gaming/Editing \n                    Enter number

        if CPUDelete==1:
            file=open("C:\\Users\\Trans\\Desktop\\Python\AMDCPUGAMING.csv","r")
            Reader=csv.reader(file)
            L=[]
            Uroll=int(input("\nEnter The SR.no. of Product to Delete.\n"))
            Found=False
            for row in Reader:
                if row[0]==str(Uroll):
                    Found=True
                else:
                    L.append(row)
            file.close()

            if Found==False:
                print('Sr.No. of Product not found')
            else:
             file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUEDITING.csv","w+",newline=")
            Writer=csv.writer(file)
            Writer.writerows(L)
            file.seek(0)
            Reader=csv.reader(file)
            for row in Reader:
                print(row)
            file.close()
```



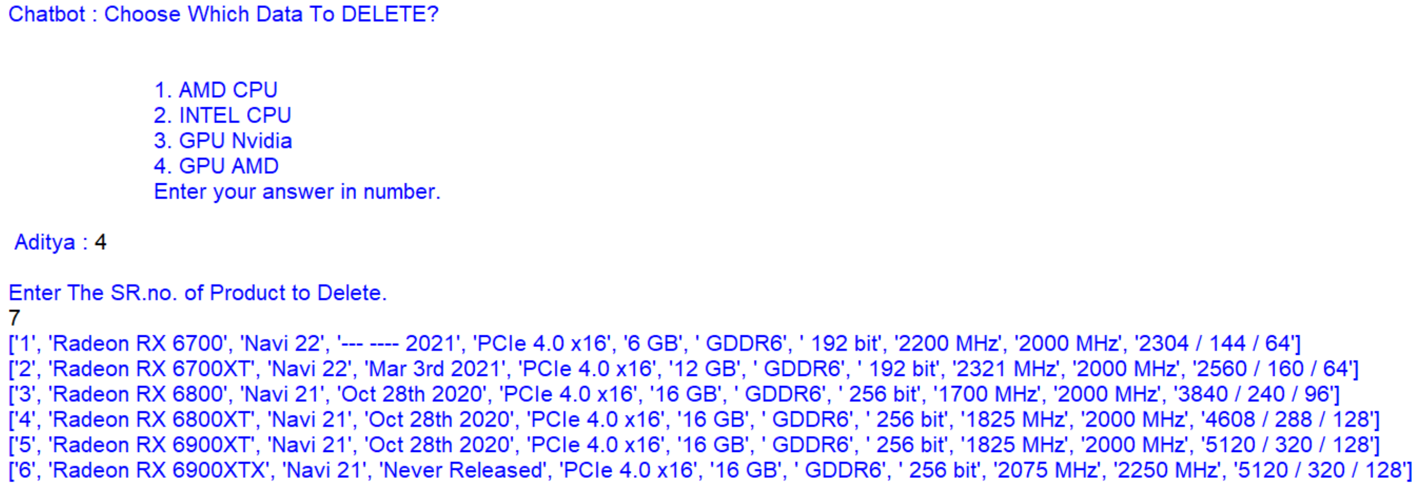
```
if CPUDelete==2:
    file=open("C:\\Users\\Trans\\Desktop\\Python\AMDCPUGAMING.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to Delete.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
        else:
            L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
     file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUEDITING.csv","w+",newline=")
    Writer=csv.writer(file)
    Writer.writerows(L)
    file.seek(0)
    Reader=csv.reader(file)
    for row in Reader:
        print(row)
    file.close()
```

```python
if DataDelete==2:
    CPUDelete=int(input("\nChatbot : WHAT IS YOUR INTEL CPU CATEGORY?\n                    1. Ligh

    if CPUDelete==1:
        file=open("C:\\Users\\Trans\\Desktop\\Python\INTELCPUGAMING.csv","r")
        Reader=csv.reader(file)
        L=[]
        Uroll=int(input("\nEnter The SR.no. of Product to Delete.\n"))
        Found=False
        for row in Reader:
            if row[0]==str(Uroll):
                Found=True
            else:
                L.append(row)
        file.close()

        if Found==False:
            print('Sr.No. of Product not found')
        else:
         file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUGAMING.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()

    if CPUDelete==2:
        file=open("C:\\Users\\Trans\\Desktop\\Python\INTELCPUEDITING.csv","r")
        Reader=csv.reader(file)
        L=[]
        Uroll=int(input("\nEnter The SR.no. of Product to Delete.\n"))
        Found=False
        for row in Reader:
            if row[0]==str(Uroll):
                Found=True
            else:
                L.append(row)
        file.close()
        if Found==False:
            print('Sr.No. of Product not found')
        else:
         file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUEDITING.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()

if DataDelete==3:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\NVIDIAGPU.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to Delete.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
        else:
            L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\NVIDIAGPU.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
```



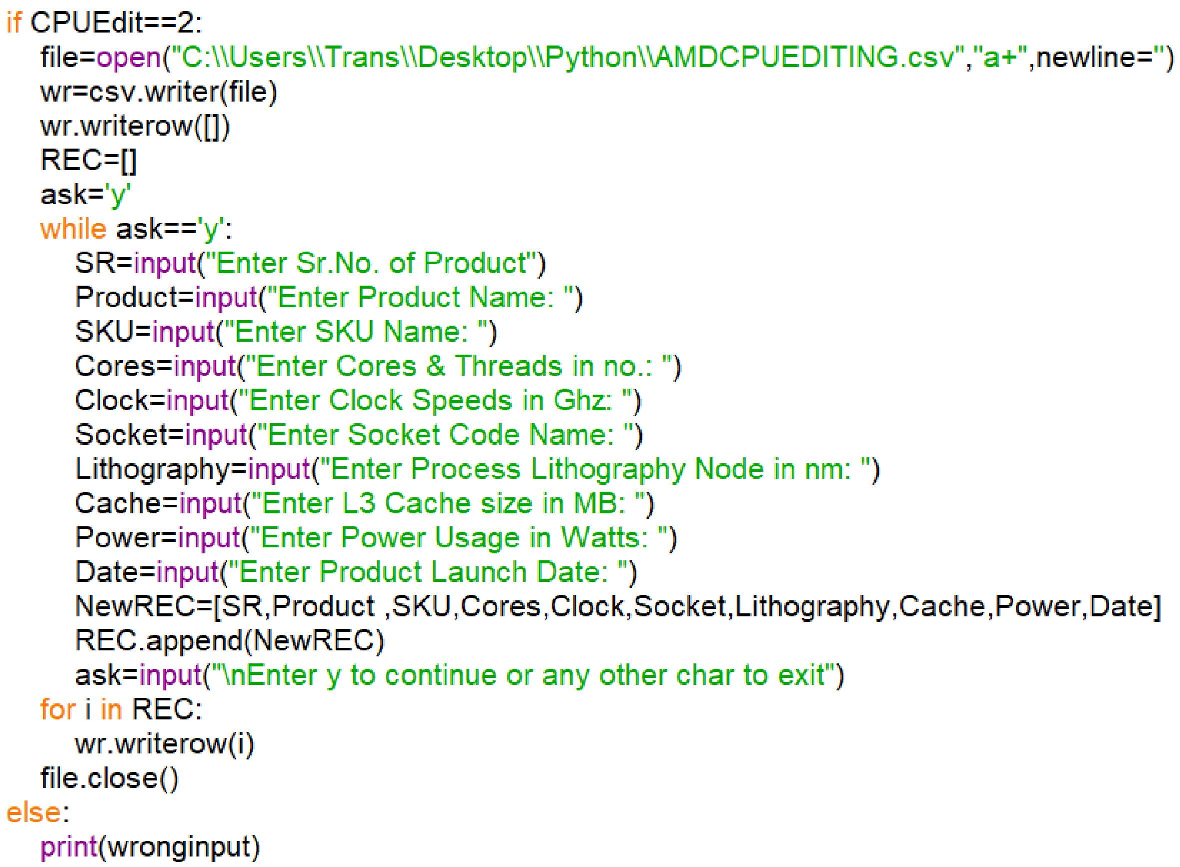
```python
if DataDelete==4:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDGPU.csv","r")
    Reader=csv.reader(file)
    L=[]
    Uroll=int(input("\nEnter The SR.no. of Product to Delete.\n"))
    Found=False
    for row in Reader:
        if row[0]==str(Uroll):
            Found=True
        else:
            L.append(row)
    file.close()

    if Found==False:
        print('Sr.No. of Product not found')
    else:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDGPU.csv","w+",newline='')
        Writer=csv.writer(file)
        Writer.writerows(L)
        file.seek(0)
        Reader=csv.reader(file)
        for row in Reader:
            print(row)
        file.close()
else:
    print(wronginput)
```

Output - Deleting One Entry Row from AMD GPU CSV:

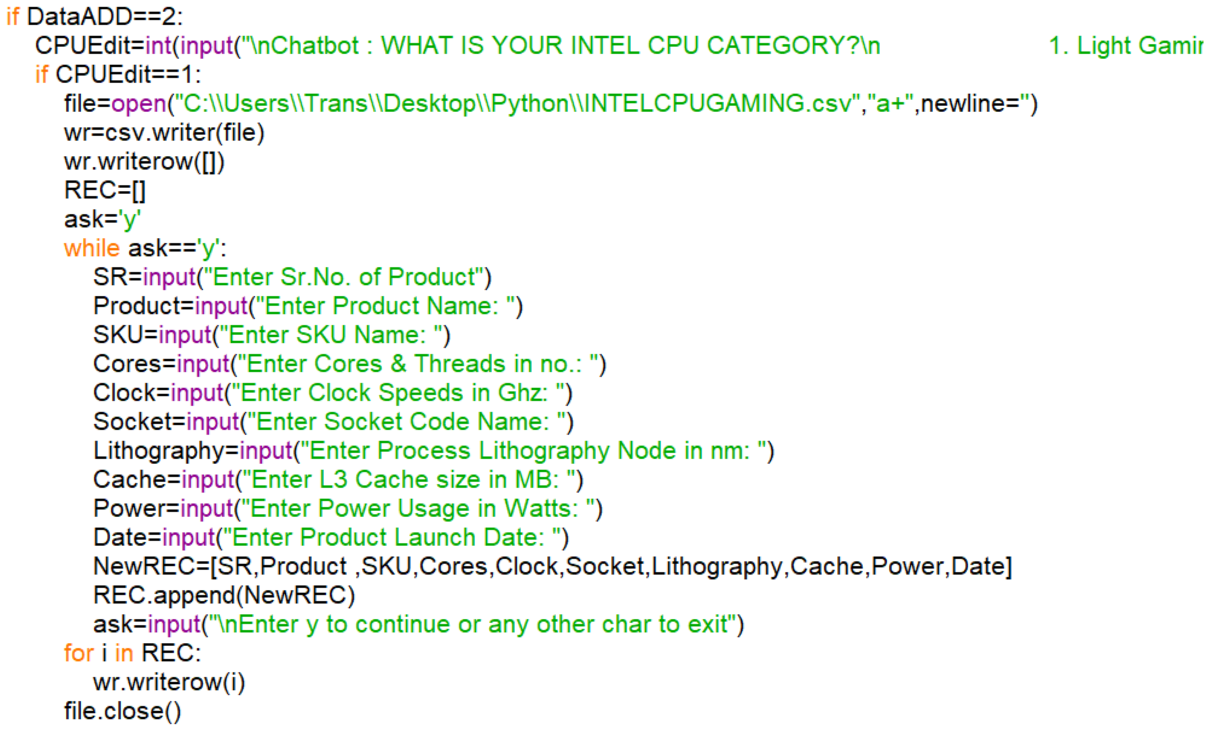


```
Chatbot : Choose Which Data To DELETE?


                    1. AMD CPU
                    2. INTEL CPU
                    3. GPU Nvidia
                    4. GPU AMD
                    Enter your answer in number.

 Aditya : 4

Enter The SR.no. of Product to Delete.
7
['1', 'Radeon RX 6700', 'Navi 22', '--- ---- 2021', 'PCIe 4.0 x16', '6 GB', ' GDDR6', ' 192 bit', '2200 MHz', '2000 MHz', '2304 / 144 / 64']
['2', 'Radeon RX 6700XT', 'Navi 22', 'Mar 3rd 2021', 'PCIe 4.0 x16', '12 GB', ' GDDR6', ' 192 bit', '2321 MHz', '2000 MHz', '2560 / 160 / 64']
['3', 'Radeon RX 6800', 'Navi 21', 'Oct 28th 2020', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '1700 MHz', '2000 MHz', '3840 / 240 / 96']
['4', 'Radeon RX 6800XT', 'Navi 21', 'Oct 28th 2020', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '1825 MHz', '2000 MHz', '4608 / 288 / 128']
['5', 'Radeon RX 6900XT', 'Navi 21', 'Oct 28th 2020', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '1825 MHz', '2000 MHz', '5120 / 320 / 128']
['6', 'Radeon RX 6900XTX', 'Navi 21', 'Never Released', 'PCIe 4.0 x16', '16 GB', ' GDDR6', ' 256 bit', '2075 MHz', '2250 MHz', '5120 / 320 / 128']
```

Choice 6 - Where to Add more details in any CSV:

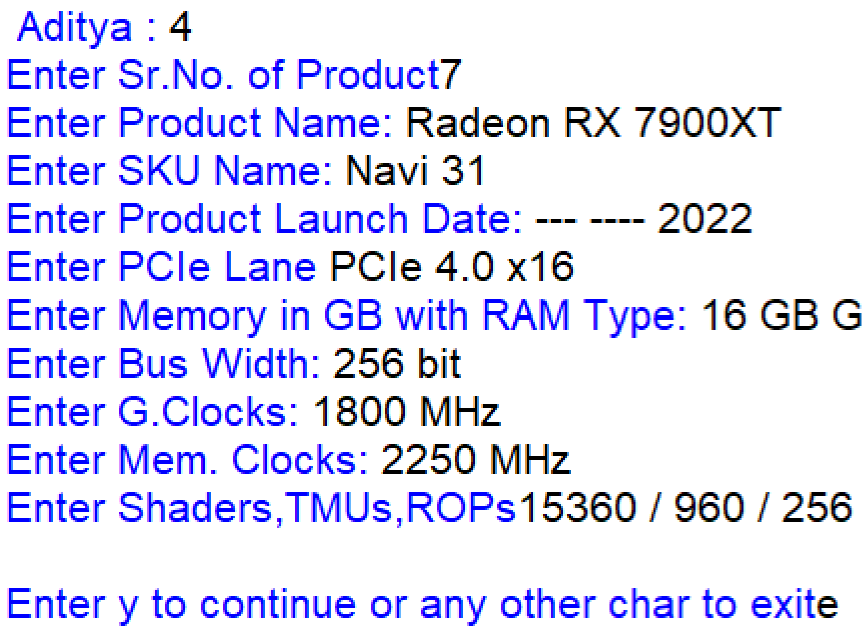


```
if CPUEdit==2:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDCPUEDITING.csv","a+",newline='')
    wr=csv.writer(file)
    wr.writerow([])
    REC=[]
    ask='y'
    while ask=='y':
        SR=input("Enter Sr.No. of Product")
        Product=input("Enter Product Name: ")
        SKU=input("Enter SKU Name: ")
        Cores=input("Enter Cores & Threads in no.: ")
        Clock=input("Enter Clock Speeds in Ghz: ")
        Socket=input("Enter Socket Code Name: ")
        Lithography=input("Enter Process Lithography Node in nm: ")
        Cache=input("Enter L3 Cache size in MB: ")
        Power=input("Enter Power Usage in Watts: ")
        Date=input("Enter Product Launch Date: ")
        NewREC=[SR,Product ,SKU,Cores,Clock,Socket,Lithography,Cache,Power,Date]
        REC.append(NewREC)
        ask=input("\nEnter y to continue or any other char to exit")
    for i in REC:
        wr.writerow(i)
    file.close()
else:
    print(wronginput)
```

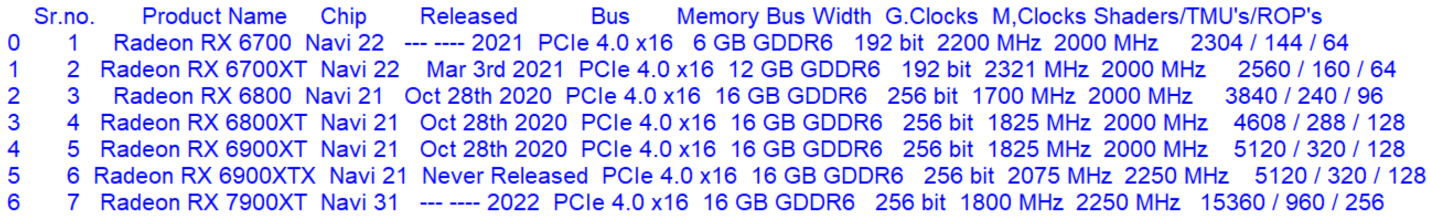


```
if DataADD==2:
    CPUEdit=int(input("\nChatbot : WHAT IS YOUR INTEL CPU CATEGORY?\n                    1. Light Gamir
    if CPUEdit==1:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUGAMING.csv","a+",newline='')
        wr=csv.writer(file)
        wr.writerow([])
        REC=[]
        ask='y'
        while ask=='y':
            SR=input("Enter Sr.No. of Product")
            Product=input("Enter Product Name: ")
            SKU=input("Enter SKU Name: ")
            Cores=input("Enter Cores & Threads in no.: ")
            Clock=input("Enter Clock Speeds in Ghz: ")
            Socket=input("Enter Socket Code Name: ")
            Lithography=input("Enter Process Lithography Node in nm: ")
            Cache=input("Enter L3 Cache size in MB: ")
            Power=input("Enter Power Usage in Watts: ")
            Date=input("Enter Product Launch Date: ")
            NewREC=[SR,Product ,SKU,Cores,Clock,Socket,Lithography,Cache,Power,Date]
            REC.append(NewREC)
            ask=input("\nEnter y to continue or any other char to exit")
        for i in REC:
            wr.writerow(i)
        file.close()
    if CPUEdit==2:
        file=open("C:\\Users\\Trans\\Desktop\\Python\\INTELCPUEDITING.csv","a+",newline='')
        wr=csv.writer(file)
        wr.writerow([])
        REC=[]
        ask='y'
        while ask=='y':
            SR=input("Enter Sr.No. of Product")
            Product=input("Enter Product Name: ")
            SKU=input("Enter SKU Name: ")
            Cores=input("Enter Cores & Threads in no.: ")
            Clock=input("Enter Clock Speeds in Ghz: ")
            Socket=input("Enter Socket Code Name: ")
            Lithography=input("Enter Process Lithography Node in nm: ")
            Cache=input("Enter L3 Cache size in MB: ")
            Power=input("Enter Power Usage in Watts: ")
            Date=input("Enter Product Launch Date: ")
            NewREC=[SR,Product ,SKU,Cores,Clock,Socket,Lithography,Cache,Power,Date]
            REC.append(NewREC)
            ask=input("\nEnter y to continue or any other char to exit")
        for i in REC:
            wr.writerow(i)
        file.close()

    else:
        print(wronginput)

if DataADD==3:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\NVIDIAGPU.csv","a+",newline='')
    wr=csv.writer(file)
    wr.writerow([])
    REC=[]
    ask='y'
    while ask=='y':
        SR=input("Enter Sr.No. of Product")
        Product=input("Enter Product Name: ")
        SKU=input("Enter SKU Name: ")
        Date=input("Enter Product Launch Date: ")
        Cores=input("Enter PCIe Lane ")
        Clock=input("Enter Memory in GB: ")
        Socket=input("Enter RAM Type: ")
        Lithography=input("Enter Bus Width: ")
        Cache=input("Enter G.Clocks: ")
        Power=input("Enter Mem. Clocks: ")
        Tflops=input("Enter Shaders,TMUs,ROPs")
        NewREC=[SR,Product ,SKU,Date,Cores,Clock,Socket,Lithography,Cache,Power,Tflops]
        REC.append(NewREC)
        ask=input("\nEnter y to continue or any other char to exit")
    for i in REC:
        wr.writerow(i)
    file.close()

if DataADD==4:
    file=open("C:\\Users\\Trans\\Desktop\\Python\\AMDGPU.csv","a+",newline='')
    wr=csv.writer(file)
    wr.writerow([])
    REC=[]
    ask='y'
    while ask=='y':
        SR=input("Enter Sr.No. of Product")
        Product=input("Enter Product Name: ")
        SKU=input("Enter SKU Name: ")
        Date=input("Enter Product Launch Date: ")
        Cores=input("Enter PCIe Lane ")
        Clock=input("Enter Memory in GB: ")
        Socket=input("Enter RAM Type: ")
        Lithography=input("Enter Bus Width: ")
        Cache=input("Enter G.Clocks: ")
        Power=input("Enter Mem. Clocks: ")
        Tflops=input("Enter Shaders,TMUs,ROPs")
        NewREC=[SR,Product ,SKU,Date,Cores,Clock,Socket,Lithography,Cache,Power,Tflops]
        REC.append(NewREC)
        ask=input("\nEnter y to continue or any other char to exit")
    for i in REC:
        wr.writerow(i)
    file.close()
```

Output Example - Add a row of Details of Component into any CSV:



```
 Aditya : 4
Enter Sr.No. of Product7
Enter Product Name: Radeon RX 7900XT
Enter SKU Name: Navi 31
Enter Product Launch Date: --- ---- 2022
Enter PCIe Lane PCIe 4.0 x16
Enter Memory in GB with RAM Type: 16 GB GDDR6
Enter Bus Width: 256 bit
Enter G.Clocks: 1800 MHz
Enter Mem. Clocks: 2250 MHz
Enter Shaders,TMUs,ROPs15360 / 960 / 256

Enter y to continue or any other char to exite
```



```
   Sr.no.      Product Name     Chip       Released          Bus       Memory Bus Width  G.Clocks  M,Clocks Shaders/TMU's/ROP's
0      1    Radeon RX 6700  Navi 22   --- ---- 2021  PCIe 4.0 x16   6 GB GDDR6   192 bit  2200 MHz  2000 MHz     2304 / 144 / 64
1      2  Radeon RX 6700XT  Navi 22    Mar 3rd 2021  PCIe 4.0 x16  12 GB GDDR6   192 bit  2321 MHz  2000 MHz     2560 / 160 / 64
2      3    Radeon RX 6800  Navi 21   Oct 28th 2020  PCIe 4.0 x16  16 GB GDDR6   256 bit  1700 MHz  2000 MHz     3840 / 240 / 96
3      4  Radeon RX 6800XT  Navi 21   Oct 28th 2020  PCIe 4.0 x16  16 GB GDDR6   256 bit  1825 MHz  2000 MHz    4608 / 288 / 128
4      5  Radeon RX 6900XT  Navi 21   Oct 28th 2020  PCIe 4.0 x16  16 GB GDDR6   256 bit  1825 MHz  2000 MHz    5120 / 320 / 128
5      6 Radeon RX 6900XTX  Navi 21  Never Released  PCIe 4.0 x16  16 GB GDDR6   256 bit  2075 MHz  2250 MHz    5120 / 320 / 128
6      7  Radeon RX 7900XT  Navi 31   --- ---- 2022  PCIe 4.0 x16  16 GB GDDR6   256 bit  1800 MHz  2250 MHz   15360 / 960 / 256
```

CONCLUSION

The project has been created with the purpose of providing the consumers with an application that will ease their work & save research time and will enable them to keep an accurate record of all important details like components' name, power, clocks, core counts & Date of Launch etc. This project is also helpful for understanding the uses, benefits and ways of using Python, Pandas and CSV files.

BIBLIOGRAPHY

•Website

https://www.techpowerup.com/

https://www.knowasiak.com/

https://geeksforgeeks.com/

•Books

IP Textbook (class 11 & class 12)