# Worksheet 9c – Two-dimensional lists Answers

1. **Creating a list to sort**

Create the following 2D list (leaving out the header row):

**cpu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CPU | Clock Speed (GHz) | Cores | Cache Size (MB) | Price (£) |
| Intel i5 6400 | 2.7 | 4 | 6 | 190 |
| Intel i7 2600 | 3.4 | 4 | 8 | 266 |
| AMD FX 8350 | 4.0 | 6 | 8 | 140 |
| AMD Ryzen 7 1700 | 3.0 | 8 | 20 | 330 |
| Intel i3 6100 | 3.7 | 2 | 3 | 110 |

cpu = [ [ “Intel i5 6400 ”, 2.7, 4, 6, 190 ] ]  
cpu.append([“Intel i7 2600 ”, 3.4, 4, 8, 266])  
cpu.append([“AMD FX 8350 ”, 4.0, 6, 8, 140])  
cpu.append([“AMD Ryzen 7 1700”, 3.0, 8, 20, 330])  
cpu.append([“Intel i3 6100 ”, 3.7, 2, 3, 110])

1. **Sorting with a lambda function**

Use a lambda function to sort the list into increasing order of price

Syntax for sorting with a lambda function:

outputList = sorted(inputList, key=lambda data:data[column])

cpuSorted = sorted(cpu, key=lambda data:data[4])  
print(cpuSorted)

1. **Sorting in reverse order**

Use a lambda function to sort the list so that the CPU with the fastest clock speed is at the top.  
  
outputList = sorted(inputList, key=lambda data:data[column], reverse=True)

cpuSorted = sorted(cpu, key=lambda data:data[1], reverse=True)  
print(cpuSorted)

Extension: Create a menu system that will ask the user which value is most important to them – name, clock speed, number of cores, cache size or price. The system should display, one line at a time and neatly, the CPUs in the appropriate order. Think about whether each one should be ascending or descending order.

print(“Choose from the following list”)  
print(“1. Name”)  
print(“2. Clock speed”)  
print(“3. Number of cores”)  
print(“4. Cache size”)  
print(“5. Price”)  
choice = int(input(“Enter your choice: ”))  
if choice == 1 or choice == 5:  
 desc = False  
else:  
 desc = True  
cpuSorted = sorted(cpu, key=lambda data:data[choice-1], reverse=desc)  
for row in range(5):  
 print(cpuSorted[row][0] + “: ” + str(cpuSorted[row][1]) + “GHz, ” + \   
 str(cpuSorted[row][2]) + “ cores ” + str(cpuSorted[row][3]) +

“MB at GBP” + str(cpuSorted[row][4]))

NB: This is just one possible solution.

The ‘\’ character allows the programmer to continue the same line of code over multiple lines in the editor. It is not needed unless the line is split between quote marks, (e.g. a long string) so is not really needed here but will not do any harm!

***See program L9 WS9c Ex3.py***