# Worksheet 9c – Two-dimensional lists

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])

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)

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.