Linear searching in python

Linear searching.py - D:\Computing 02.12.16\Mrs Berry\Python Programming\recent\Using arrays\Linear searching.py

File Edit Format Run Options Windows Help

```
#Linear searching
list01 = ["aaples", "bananas", "choc", "pasta"] #global list

def main_linear_proc(): #Not using functions
    print("Items in the list are: ", list01)
    search_item = str(input("Enter name of item you want to find: "))
    found = False #Boolean value
    position = 0

    while (position < len(list01)) and (not found): #not found just means found = false
        if list01[position] == search_item:
            found = True
            print("Found item at position", position, "\n")
        else:
            position = position + 1
    if not found:
        print("Item not found in list\n")

main_linear_proc()
```

Using functions for the same algorithm

```
def main_linear_proc2(): #using functions
    def func_search(search_item,list01):
        found = False                           #Boolean value
        position = 0                            #starting position of list

        while (position < len(list01)) and (list01[position] != search_item): #i.e. not found
            position = position + 1             # Item is not found, position adds 1
        if position < len(list01):
            found = True                        #item found, so found changes to True

        return (found,position)                 #returns two values: state of found (T/F) & position as list

    ###Setup
    print("\tUsing functions: \n")
    print("\nItems in the list are: ", list01)
    search_item = str(input("Enter name of item you want to find: "))
    (found,position) = func_search(search_item,list01) ##run func to get two values

    if found == True:
        print("Item has been identified in the list")
        print("Item is at index position: ", position) #taking 0 as start
    else:
        print("The item was never found")

#loops program
response = "Y"
while response == "Y":
    main_linear_proc2()
    response = str(input("\nContinue? (y/n)"))
    response = response.upper()
```

Linear searching will start at the first element and compare each element in the list with the target value. Once the item is found the iteration stops and the position stated, if the end of the list is reached without the target value being identified then the search stops.

Linear searching is useful for searching a list with a small number of elements or a list that only has to be searched once. It is considered less efficient than binary searching since every element in the list must be checked unit the item found. In the worst case scenario, the item may be the last element in the list so there are many unnecessary comparisons/ iterations made. A big advantage of linear searching is that it works even when the list is not ordered since every element will be checked regardless of the order, until the item is found or end of list is reached.

Binary searching in python

```
#Binary searching

def func_binary_search(search_item,list01):
    lower = 0                          #equals first position in list
    upper = (len(list01) - 1)         # since we want last position not full length (it begins at 0)
    found = False

    while (found == False) and (lower <= upper):
        mid = int((lower+upper)//2) # the middle position (full length-1 divided by 2 (same as a%b whole integer division)
        if list01[mid] == search_item:
            found = True            # ITEM FOUND, breaks the while

        elif list01[mid] > search_item:
            upper = (mid - 1)           #the end of the list is now element below middle item
        else:                           # list01[mid] < search_item:
            lower = mid + 1

    return found #(T/F)

def main_binary_search(): #using functions

    ###Setup
    list01 = [1,2,3,4,5,6,7,8,9,20,34,40,45,46,47,48,50,70,90,100] #ordered integer list
    print("\nItems in the list are: ", list01)
    search_item = int(input("Enter the integer of the item you want to find: "))

    if (search_item > list01[len(list01)-1]) or (search_item < list01[0]): #item not in range of list at all
        print("The integer entered is not within range of list") #we immediately know it is not in list
        found = False
    else:
        found = func_binary_search(search_item,list01) ##run function to obtain position

    print("\nItem in list:", found)

#loops program
response = "Y"
while response == "Y":
    main_binary_search()
    response = str(input("\nContinue? (y/n)"))
    response = response.upper()
```

Binary search is an efficient searching algorithm for finding an item within an ordered list. It works by repeatedly comparing the middle item of the list with the target value, and if it is not equal, the list is divided in half. The portion of the list that could contain the value is amended as the new list. The process repeats until the possible location of the item is narrowed down to one. This is then compared to target value.