

Assignment 3: Programming in Python CST 362

Name: **Alby Thekkedan** Roll No **8** Class **CS6B**

KTU-ID: **MDL20CS129**

Date of submission: 15-March-2023

Learning outcome: Learn list, tuple and set

1. Write commands for the following
 - a. Create an empty list (stud)
 - b. Add your roll number to the list (use + operator)
 - c. Append your name (use append)
 - d. Extend the list with your place, pin, number (extend)
 - e. Insert your KTU-ID at stud [0]
 - f. Print your KTU-ID and name
 - g. Print the number of characters in your name.
 - h. Print last 5 digit of the phone number
 - i. Reverse the stud list
 - j. Find the index of your name.

```
8
8 Alby
8 Alby Vadavucode 682310 9747554404
MDL20CS129 8 Alby Vadavucode 682310 9747554404
MDL20CS129 Alby
4
54404
9747554404 682310 Vadavucode Alby 8 MDL20CS129
9747554404 682310 Vadavucode Alby 8 MDL20CS129
3
```

```
stud=[]

stud+=[8]
print(*stud)

stud.append('Alby')
print(*stud)

stud.extend(['Vadavucode',682310,9747554404])
print(*stud)

#stud[0]='MDL20CS129'
#print(stud)

stud=['MDL20CS129']+stud
print(*stud)

print(stud[0], stud[2])

print(len(stud[2]))

phno=str(stud[5])
last4dig=phno[-5:]
print(last4dig)

print(*stud[::-1]) # Just prints reverse
stud.reverse()
print(*stud)

name='Alby'
IndexOfName= stud.index(name)
print(IndexOfName)
```

2. Read list of numbers and store it in a list. Create two new lists from the list created which contains prime and composite numbers.

```
Enter the number of numbers5
Enter number 11
Enter number 1
Enter number 2
Enter number 9
Enter number 7
Original List:  11 1 2 9 7
Prime numbers:  11 2 7
Composite numbers:  1 9
```

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

num_list = []
count=int(input("Enter the number of numbers"))
cnt=0
while (cnt!=count):
    num = int(input("Enter number "))
    num_list.append(num)
    cnt+=1

prime_list = []
composite_list = []
for num in num_list:
    if is_prime(num):
        prime_list.append(num)
    else:
        composite_list.append(num)
print("Original List: ", *num_list)
print("Prime numbers: ", *prime_list)
print("Composite numbers: ", *composite_list)
```

3. Read list of students names and do the following
 - a) Sort the name in alphabetical order
 - b) Find the name with largest length
 - c) print the names starting with letter 'A' (assume first letter capital)
 - d) Print the names in reverse alphabetical order with all names converted to capital letters.
 - e) Print the names in the order of length

```
names=['Sumegh','Alby','abhijith','akash','Amal']

# Sort the names in alphabetical order
sorted_names = sorted(names)

# Find the name with Largest Length
longest_name = max(names, key=len)

# Print the names starting with Letter 'A'
A_names = [name for name in names if name.startswith('A')]

# Print the names in reverse alphabetical order with all names converted to capital letters
reversed_names = sorted(names, reverse=True)
capitalized_names = [name.upper() for name in reversed_names]

# Print the names in the order of Length
length_sorted_names = sorted(names, key=len)

print("Sorted names:", *sorted_names)
print("Longest name:", *longest_name)
print("Names starting with 'A':", *A_names)
print("Names in reverse alphabetical order:", *capitalized_names)
print("Names sorted by length:", *length_sorted_names)
```

Sorted names: Alby Amal Sumegh abhijith akash
 Longest name: a b h i j i t h
 Names starting with 'A': Alby Amal
 Names in reverse alphabetical order: AKASH ABHIJITH SUMEGH AMAL ALBY
 Names sorted by length: Alby Amal akash Sumegh abhijith

```
print([3**i for i in range(21)])
print("\n")
print([i for i in range(1, 100) if i % 3 == 0])
```

4. Use list comprehension to create lists
 - a) powers of 3 upto 20
 - b) Numbers less than 100 which are divisible by 3

```
[1, 3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, 177147, 531441, 1594323, 4782969, 14348907, 43046721, 129140163, 387420489, 1162261467, 3486784401]
```

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99]
```

5. Read 10 numbers and stores it in a tuple. Find the sum, average of these elements also find the largest and smallest.

```
numbers=(2,4,6,8,10,12,14,16,18,20)
total = sum(numbers)
average = total / len(numbers)
largest = max(numbers)
smallest = min(numbers)
print(f"Tuple: ",numbers)
print(f"Sum: ",total)
print(f"Average: ",average)
print(f"Largest element: ",largest)
print(f"Smallest element: ",smallest)
```

Tuple: (2, 4, 6, 8, 10, 12, 14, 16, 18, 20)
 Sum: 110
 Average: 11.00
 Largest element: 20
 Smallest element: 2

6. Read the sets. Find the union, intersection and symmetric difference of two sets A and B.

```
A={2,4,6,8,10,12}
print("Set A:", A)
B={4,8,12,16,20}
print("Set B:", B)
print("Union ",A|B)
print("Intersection ",A&B)
print("Symmetric difference of A and B: ",A^B)
```

Set A: {2, 4, 6, 8, 10, 12}
 Set B: {16, 4, 20, 8, 12}
 Union {2, 4, 6, 8, 10, 12, 16, 20}
 Intersection {8, 4, 12}
 Symmetric difference of A and B: {16, 2, 20, 6, 10}

7. Read a string and print the words in alphabetical order.

```
string = input("Enter a string: ")
words = set(string.split())
for word in sorted(words):
    print(word)
```

Enter a string: an apple a day keeps the doctor away
a
an
apple
away
day
doctor
keeps
the

8. Read a string and print the words and its length. Also find the average word length.

```
string = input("Enter a string: ")
words = string.split()
for word in words:
    print("The length of ",word,"is ",(len(word)))
unique_lengths = set(len(word) for word in words)
average_length = sum(unique_lengths) / len(unique_lengths)
print("Average word length: ",average_length)
```

Enter a string: I love Model Engineering
The length of I is 1
The length of love is 4
The length of Model is 5
The length of Engineering is 11
Average word length: 5.25

9. Read list of numbers and find the mean, median and standard deviation.

```
numbers=[2,4,6,8,10,12,14,16,18,20]
n=len(numbers)
mean = sum(numbers) / n

sorted_numbers = sorted(numbers)
if n % 2 == 0:
    median = (sorted_numbers[n // 2 - 1] + sorted_numbers[n // 2]) / 2
else:
    median = sorted_numbers[n // 2]

sum_squares = sum((x - mean) ** 2 for x in numbers)
std_dev = (sum_squares / n) ** 0.5

print("Mean: ",mean)
print("Median: ",median)
print("Standard deviation: ",std_dev)
```

Mean: 11.0
Median: 11.0
Standard deviation: 5.744562646538029

10. Consider a list consisting of integers, floating point numbers and strings. Separate them into different lists depending on the data types.

```
A = [1, 2.0, "three", 4, 5.5, "six", 7, 8.8, "nine"]
```

```
int_list = []
float_list = []
str_list = []

for item in A:
    if type(item) == int:
        int_list.append(item)
    elif type(item) == float:
        float_list.append(item)
    elif type(item) == str:
        str_list.append(item)
print("Integers:", int_list)
print("Floating-point numbers:", float_list)
print("Strings:", str_list)
```

Integers: [1, 4, 7]
Floating-point numbers: [2.0, 5.5, 8.8]
Strings: ['three', 'six', 'nine']

11. Check if the items in the list are sorted in ascending or descending order and print suitable messages accordingly. Otherwise, print "Items in list are not sorted"

```
A = [1, 2, 3, 4, 5]

if A == sorted(A):
    print("Items sorted in ascending order")

elif A == sorted(A, reverse=True):
    print("Items sorted in descending order")

else:
    print("Items not sorted")
```

Items sorted in ascending order

12. Remove all duplicate elements from a list

```
A = [10, 20, 20, 30, 40]
B = set(A)
print("Before ", *A)
print("After Duplicates removed:", *B)
```

Before 10 20 20 30 40

After Duplicates removed: 40 10 20 30

13. Find the number with largest frequency of occurrence. i/p:10 20 30 40 40 40 50 50 o/p:40 (Note: there may be more than one element)

```
A = [10, 20, 30, 40, 40, 40, 50, 50, 60, 60, 60]
max_freq = 0
max_nums = []

for num in A:
    freq = A.count(num)

    if freq > max_freq:
        max_freq = freq
        max_nums = [num]

    elif freq == max_freq:
        max_nums.append(num)
print("Number(s) with the maximum frequency:", set(max_nums))
```

Number(s) with the maximum frequency: {40, 60}