***CCCS 104 - Data Structures and Algorithms***

LEARNING TASK (LINEAR DATA STRUCTURE - LIST)

GROUP NO: 9 SECTION: BSCS 2A

GROUP LEADER: Venn P. Delos Santos

GROUP MEMBERS: Marc Christian D. Tumaneng

John Mark A. Pajenago

**RATIONALE**

|  |
| --- |
| *Explain briefly List Linear Data Structure, how it works? What are the common examples? Its applications?*   * *Data structures such as List Linear Data Structure is a specific way of organizing data in a specialized format on a computer so that the information can be organized, processed, stored, and retrieved quickly and effectively. A linked list is a linear data structure where each element is a separate object. It is made up of two items that are data and a reference to the next node. A reference to the next node is given with the help of pointers and data is the value of a node. Each node contains data and links to the other nodes. It is an ordered collection of data elements called a node and the linear order is maintained by pointers. List Linear Data Structure are mostly utilized in software development and there are application for list linear data structure which are the implementation of stack and queue, implementation of graph, dynamic memory allocation, performing arithmetic operations on long integers and many more.*   *Introduce your develop Python program, what can it do?*   * *The Python Program that we developed implements List Data Structure using Linked List. It also has a Menu that has many choices that are going to perform data structures operations like insertion, deletion, and search. The program also asks for user input such as the value to be inserted, and the position or the index of a given integer. It is also capable of looping in which the menu’s choices are possible to be selected and the user’s inputs are possible to be stored in every loop.* |

**USER GUIDE**

*Step by step instructions on how to use your program. Include images for easily visualization*

|  |
| --- |
| *Step 1*  ***Enter the user input from 1-10*** |
| *Image 1* |
|  |
| *Step 2*  ***-Select [1] to insert at start by typing 1***  ***-input the integer to be inserted.*** |
| *Image 2* |
| *Step 3*  ***-Select [2] to insert at end by typing 2***  ***-input the integer to be inserted*** |
| *Image 3* |
|  |
| *Step 4*  ***-Select [3] to insert at position by typing 3***  ***-input the integer to be inserted***  ***-input the position where to insert the integer***  ***\*note***  ***-to check all the integer in the list, select [10]*** |
| *Image 4*    *\*note: to check the integers’ positions* ***[10]*** |
|  |
|  |
|  |
|  |
| *Step 5*  ***-Select [4] to delete at start by typing 4*** |
| *Image 5* |
|  |
| *Step 6*  ***-Select [5] to delete at end by typing 5*** |
| *Image 6* |
|  |
| *Step 7*  ***-Select [6] to delete at position by typing 6***  ***-enter the position of the value to be deleted*** |
| *Image 7* |
|  |
|  |
|  |
| *Step 8*  ***-Select [7] to delete at position by typing 7***  ***- enter the value of the number to be deleted*** |
| *Image 8* |
|  |
| *Step 9*  ***-Select [8] to delete at position by typing 8***  ***-enter the value to search*** |
| *Image 9* |
|  |
| *Step 10*  ***-Select [9] to display number at position by typing 9***  ***-enter the position of the value*** |
| *Image 10* |
|  |
|  |
| *Step 11*  ***Select [10] to display linked list by typing 10*** |
| *Image 11*  *\*After inserting 5 integers in the list using [1] Insert at start (2, 4, 5), [2] Insert at end (1), and [3] Insert at position (3 at index 2).*    *\*After deleting 4 Integers using choices [4]**Delete at Start, [5] Delete at End,*  *[6] Delete at position, [7] Delete Number.* |
|  |
| *Step 12*  ***Select [0] to exit by typing 0*** |
| *Image 12* |

**PROGRAM CODE**

|  |
| --- |
| *Insert your Python code here, be sure to add comments to describe each line/set of codes…* |

#Python program that will implement a List Data Structure using Linked List and menu

#define class node

class Node:

  def \_\_init\_\_(self,data=None,next=None):

    self.data=data

    self.next=next

  def print(self):

    pass

#define class

class LinkedList:

  def \_\_init\_\_(self):

    self.head=None

  # Insert At Start

  def push\_front(self,data):

    self.head=Node(data,self.head)

  # Insert At End

  def push\_back(self,data):

    i=self.head

    if i:

      while i.next:

        i=i.next

      i.next=Node(data,i.next)

    else:

      self.head=Node(data,self.head)

  #Insert At Position

  def push\_at(self,index,data):

    i=self.head

    if i and index>0:

      j=0

      while i.next and j<index-1:

        j+=1

        i=i.next

      i.next=Node(data,i.next)

    else:

      self.head=Node(data,self.head)

  #Delete At End

  def pop\_back(self):

    i=self.head

    if i:

      while i.next:

        j=i

        i=i.next

      j.next=i.next

    return i

  #Delete At Position

  def pop\_at(self,index):

    i=self.head

    if i:

      if index>0:

        k=0

        while i.next and k<index:

          k+=1

          j=i

          i=i.next

        j.next=i.next

      elif index==0:

        self.head=self.head.next

    return i

  #Delete At Start

  def pop\_front(self):

    node=self.head

    self.head=self.head.next

    return node

  #Display Number At Position

  def peek\_at(self,index):

    i=self.head

    if i and index>0:

        j=0

        while i.next and j<index:

          j+=1

          i=i.next

    return i

  #Search Number

  def search(self,data):

    i=self.head

    j=-1

    k=0

    while i:

      if i.data==data:

        j=k

        break

      k+=1

      i=i.next

    return j

  # Display items of the List

  def print(self):

    i=self.head

    print('head',end='->')

    while i:

      print(i.data,end='->')

      i=i.next

    print('None')

#define class

class LinkedListSystem:

  def \_\_init\_\_(self):

    self.linked\_list=LinkedList()

  #define menu

  def main\_menu(self):

    quit=False

    while not quit:

     # display all the choices in the menu

      print(

        '\nLinkedList Main Menu:\n'+

        '[1] Insert at Start\n'+

        '[2] Insert at End\n'+

        '[3] Insert at Position\n'+

        '[4] Delete at Start\n'+

        '[5] Delete at End\n'+

        '[6] Delete at Position\n'+

        '[7] Delete Number\n'+

        '[8] Search Number\n'+

        '[9] Display Number At Position\n'+

        '[10] Display Linked List\n'+

        '[0] Exit\n\n'

      )

     # ask for the user input

      choice=int(input('Enter choice: '))

      #if else statement that assigns function to the menu choices

      if choice==0:

        quit=True

        continue

      elif choice==1:

        self.insert\_at\_start()

      elif choice==2:

        self.insert\_at\_end()

      elif choice==3:

        self.insert\_at\_position()

      elif choice==4:

        self.delete\_at\_start()

      elif choice==5:

        self.delete\_at\_end()

      elif choice==6:

        self.delete\_at\_position()

      elif choice==7:

        self.delete\_number()

      elif choice==8:

        self.search\_number()

      elif choice==9:

        self.display\_number\_at\_position()

      elif choice==10:

        self.print\_linked\_list()

    print('Thank you!')

  #function of choice 1

  def insert\_at\_start(self):

    data=int(input('Enter integer to insert: '))

    self.linked\_list.push\_front(data)

    print('Data is inserted at beginning')

  #function of choice 2

  def insert\_at\_end(self):

    data=int(input('Enter integer to insert: '))

    self.linked\_list.push\_back(data)

    print('Data Inserted at end')

  #function of choice 3

  def insert\_at\_position(self):

    data=int(input('Enter integer to insert: '))

    index=int(input('Insert at position: '))

    self.linked\_list.push\_at(index,data)

    print('Data Inserted at position',index)

  #function of choice 4

  def delete\_at\_start(self):

    print('Removed value from beginning: ',self.linked\_list.pop\_front().data)

  #function of choice 5

  def delete\_at\_end(self):

    print('Removed value at end: ',self.linked\_list.pop\_back().data)

  #function of choice 6

  def delete\_at\_position(self):

    index=int(input('Remove value at position: '))

    self.linked\_list.pop\_at(index)

    print('Data removed  at',index)

  #function of choice 7

  def delete\_number(self):

    data=int(input(' Enters the value of the number to be deleted: '))

    index=self.linked\_list.pop\_at(data)

    if index==-1:

      print('Value',data,'“is not on the list and cannot be deleted.”')

    else:

      print('Value',data,'has been successfully deleted.')

  #function of choice 8

  def search\_number(self):

    data=int(input('Enter value to search: '))

    index=self.linked\_list.search(data)

    if index==-1:

      print('Value',data,'not found.')

    else:

      print('Value',data,'is at position',index)

  #function of choice 9

  def display\_number\_at\_position(self):

    index=int(input('Get value at position: '))

    print('Value at position',index,'is',self.linked\_list.peek\_at(index).data)

  #function of choice 10

  def print\_linked\_list(self):

    self.linked\_list.print()

sys=LinkedListSystem()

sys.main\_menu()

**TUTORIAL VIDEO**

|  |
| --- |
| YouTube Link: <https://youtu.be/YN2oC-HJEFk> |

**TAKEAWAYS**

|  |
| --- |
| Name of Member: Venn P. Delos Santos |
| Contribution to the Group: coding the program & video documentation |
| Learnings:  Learned the idea how to perform Data Structures Operations such as Insertion, Deletion, and Search in a simple Linked List program. |

|  |
| --- |
| Name of Member: Marc Christian D. Tumaneng |
| Contribution to the Group: coding the program & video documentation |
| Learnings: I learned the concept of List Data Structure, linked list and somehow understand its applications. |

|  |
| --- |
| Name of Member: John Mark A. Pajenago |
| Contribution to the Group: coding the program & video documentation |
| Learnings: I learned that how to create node and distinguish what is linked list and how it work and to code in python |

**REFERENCES**

Simplilearn. (n.d). *What is Data Structure: Types, Classifications and Applications.*

[*https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-data-structure*](https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-data-structure)

GeeksforGeeks. (n.d). *Applications of linked list data structure.*

[*https://www.geeksforgeeks.org/applications-of-linked-list-data-structure/*](https://www.geeksforgeeks.org/applications-of-linked-list-data-structure/)

GeeksforGeeks. (n.d). *Overview of Data Structures | Set 1 (Linear Data Structures).*

<https://www.geeksforgeeks.org/overview-of-data-structures-set-1-linear-data-structures/>

El Agua F. (2022) [Programmers Codeposting]. Facebook. [Source code]. http://ix.io/49rl?fbclid=IwAR1hv2IkSoi-nQ-CjBq65W-wNl\_pZLQTTZPsfrnl7bEWS\_TWgquZI18eLsA