

**Chittagong University of Engineering and Technology**

**Title: DNA Visualization By Using Python.**

**Submitted by:**

**ID:**

1911016

1911017

1911018

**Submitted by:**

| ID      | NAME                    | REMARKS |
|---------|-------------------------|---------|
| 1911016 | MD.TANVIR HOSSAIN KANOK |         |
| 1911017 | MD.OBAYED ULLAH         |         |
| 1911018 | MD.JAHID HASAN          |         |

**DEPARTMENT : BIOMEDICAL ENGINEERING( BME)**

**COURSE TITLE : OBJECT ORIENTED PROGRAMMING SESSIONAL**

**COURSE CODE : CSE 224**

**ABSTRACT:**

The name of this project is “ DNA visualization using Python”. This project will show the structure of DNA double helix by taking random input from one hundred DNA samples. One hundred DNA samples are in a text file. In this program we use Pandas Library so that this program can read text file and collect data from the text file. Then this program will take input from users ,if this input is in the text file then it will take data from the file and show the double helix structure of DNA.

## TABLE OF CONTENTS

| Topics                | Page      |
|-----------------------|-----------|
| 1.Objective           | 05        |
| 2.Introduction        | 05        |
| 3.Motivation          | <u>05</u> |
| 4.Implementation      | 06-09     |
| 4.1.Algorithm         | 06        |
| 4.2.Flow Chart        | 07        |
| 4.3.Data structures   | 07        |
| 4.4.Functions         | 08        |
| 4.5.Special Module    | 09        |
| 5.Sample Input/Output | 10-12     |
| 6.Discussion          | 12        |
| 7.Future work         | 13        |
| 8.Conclution          | 13        |
| 9.Appendix            | 14-16     |

### **OBJECTIVE:**

- To build a system that will Visualize the DNA.

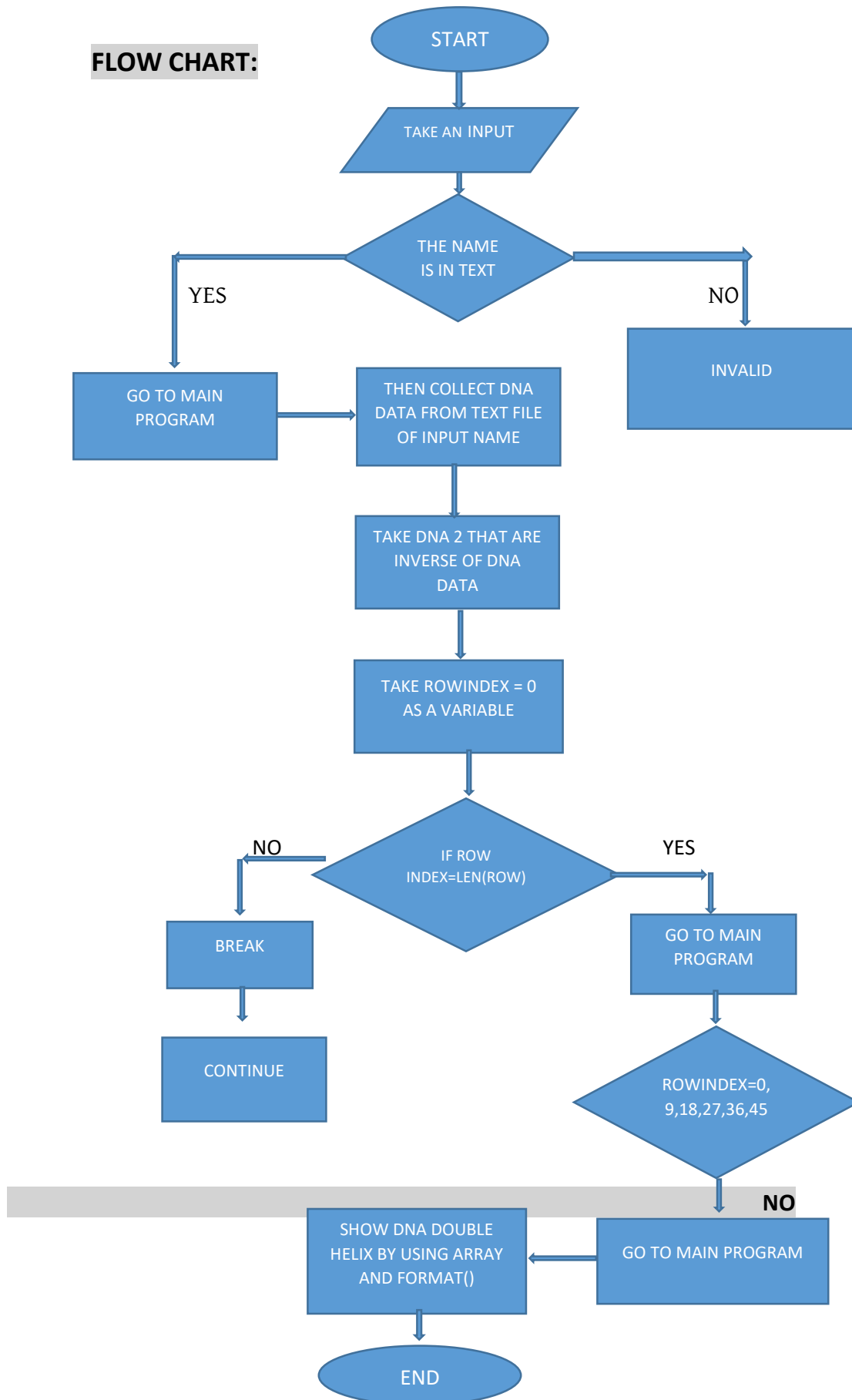
### **INTRODUCTION:**

To read the text file this program imports Pandas then shows up the text file data. Then this program will take a name from the user, if the name stays in the text file it will execute the output otherwise it will be invalid. If the program is valid, it will show the DNA double helix structure of the person. This program uses an array to form the double helix structure. Index no-9,18,27,36,45 remains empty. For inserting specific values Format () is used by using for loop. Then if we run the program, it will show the DNA structure of the specific person whose name is given as input.

### **MOTIVATION:**

- To know about the DNA structure of a person.
- To observe the differences in DNA structure of various people.
- DNA sequencing can be used in medical science for various purposes.
- Scientists can detect genome sequences.
- DNA sequencing can be used to detect the genes which are associated with some acquired diseases like cancers can be detected by observing certain genes.

**FLOW CHART:**



### **DATA STRUCTURE:**

- **LIST:** Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

**WHY WE USE LIST:** Inputs are taken from text file by list.

- **Dictionary:** Dictionaries are used to store data values in key value pairs. A dictionary is a collection which is ordered, changeable and does not allow duplicates. As for Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

**WHY WE USE DICTONAY:** Dictionary is used for collection of data by index formatting.

### **DATA STORAGE AND HANDLING:**

File handling in Python is simplified with built-in methods, which include creating, opening, and closing files. While files are open, Python additionally allows performing various file operations, such as reading, writing, and appending information.

**In this program text file are handle by this manner:**

```
Import pandas as pd
```

```
File = pd.read_csv('location of file/name of this file')
```

```
File.head()
```

In this manner this program takes data from text file.

## **FUNCTIONS:**

In programming, a function is a block of code which performs a specific task. It only runs when it is called and also can return data as a result. There were a massive number of functions that were used in the project.

1. `len ()` → this function was used to return the number of items in the

intermediate object and source object

2. `split()` → it splits the string into a list. Here we separate the folder name and file name from the source path and also split the files name into the prefix and postfix variable.

3. `append()` → this function was used to add `prefix(file names )` in the list called

`intermediate_file`

4. `range ()` → this was used to count the file number in the source object.

5. `join()` → this function was used to join the prefix and postfix name with “.”

ex: `test.py`.

Here `test` is the prefix and `py` is the postfix.

6. `open()` → it opens a file and returns it as a file object.

7. `readline()` → `readline` is a method that reads line by line

8. `input()` → `input ()` was used to get the opinion about keeping the file or not

from the user

9. `write ()` → this is used to overwrite the existing file

10. `glob()` → it returns an object, the file paths are attributes of which.



11. `call()` → it is used to run other python programs from the script.

### **SPECIAL MODULE:**

Any text file with the `.py` extension containing Python code is basically a module.

Different Python

objects such as functions, classes, variables, constants, etc., defined in one module can be made

available to an interpreter session or another Python script by using the `import` statement.

**2 modules were used in this programming, they are:**

1. GLOB MODULE : globe (short for global) module used to search file paths that match a

specific pattern. We import glob functions here.

2. SUBPROCESS MODULE : Subprocess module used to run new codes and applications by

creating new processes. We import call functions from this module.

**INPUT:**

```
In [12]: file.head()
```

|   | Patient_Name | dna_data  |
|---|--------------|---|
| 0 | Ahad         | ATGCCCCAACTAAATACCGCCGTATGACCCACCATAATTACCCCCATA  |
| 1 | Arif         | ACACTATTTCCTGTCACCCAACTAAAAATATTAAATTCAAATTACCAT  |
| 2 | Asad         | ACAATAAACCTGAGAACCAAAATGAACGAAAATCTATTCGCTTCACT   |
| 3 | Asikh        | AAAATGAACGAAAATCTATTCTGCTTCATTCTGCTGCCCCACAATCCTA |
| 4 | Hasan        | CCGTCGGGGCCGAGCGCCGCTGCCGACGTCGCGCGCAGGTTTCATAGG  |

```
dna2 = dna[ : :-1]
print(len(dna2))
import random, time
PAUSE = 0.15
ROWS = [

    '##', # Index 0 has no {}
    '#{}-{}#',
    '#{}--{}#',
    '#{}---{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '##', #9
    '#{}-{}#',
    '#{}--{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '##', #18
    '#{}-{}#',
    '#{}--{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    '#{}-----{}#',
    ]
```

```

    #{}-----{}# ,
    #{}-----{}#',
    #{}---{}#',
    #{}-{}#',
    ##', #45
    #{}-{}#',
    #{}---{}#',
    #{}-----{}#',
    #{}-----{}#',
    #{}-----{}#',
    #{}-----{}#',
    #{}---{}#',
    #{}-{}#',
    ##',

]

print('DNA data')
rowIndex = 0

for i,j in zip(dna,dna2):
    rowIndex = rowIndex + 1
    if rowIndex == len(ROWS):
        break
    # Row indexes 0 and 9 don't have nucleotides
    if rowIndex == 0 or rowIndex == 9 or rowIndex==18 or rowIndex==27 or rowIndex==36 or rowIndex==45:
        continue
    print(ROWS[rowIndex].format(i,j))

```

```

|
Patient_Name dna_data
Ahad ATGCCCCAACTAAATACCGCCGTATGACCCACCATAATTACCCCCATA
Arif ACACTATTTCTCGTCACCCAACTAAAAATATTAAATTCAAATTACCAT
Asad ACAATAAACCCCTGAGAACCAAAATGAACGAAAATCTATTCGCTTCACT
Asikh AAAATGAACGAAAATCTATTCGCTTCATTTCGCTGCCCCCACAATCCTA
Hasan CCGTCGGGGCCGAGCGCCGCTGCCGACGTCCGCGCGCAGGTTTCATAGG
Reyana TAATAACTATACACAGCACTAAAGGACGAACCTGATCTCTCATACTAGTA
Shorna CCGTCGGGGCCGAGCGCCGCTGCCGACGTCCGCGCGCAGGTTTCATAGG
Trina CGGAAGGGACTTGACTTGTCTCAGATACCCTATATTAATAAACAACCTAC
Rakib ACCTCGCCCAGCAGTACTGCCATGAAGCAATAAGAGAGATCAGTAAACT
Ibrahim GCGGCAGCGGGGGCAGTGGGTCTCGAGAGACGGGACTGGAGTCCCACT
Sadiah CCACCCGCCACGCCCCGAGCAGGGCCTGTCCGCCTTCTACCTCTCCTAGG
Neha AGTGGACCTTGCATCCCGCGAGGGCACCAGCACTCAGAACCAGCCACC
Tanim CCCCAGGGGGCAGCCTGGACTTGGTGCCCGGCGGGCTGACCTTGGAGG
Pritam AGCACTCGCTGGAGCAGGTGCAGTCCATGGTGGTGGGCGAAGTGCTCA
Saiful GGAAGTGGAGCTCCAGCAATGTGCAGAAAGTGGCTCCTGTGGACAGAGCAC
Muktadi AGGAGCCACCTGAGGAGCCTGAGCAGTGCCCGGTTCATTGACAGCCAAG
Usha TTTGGCTAGGGACTTTGCCTATGTCTGTGAAGCCGAATTTCTAGTAAAC
Arko GAGAAGTTAGACAAGATTGGGTTGAATCTTCCGGCTGGGAGGCGGAAA
Kanok GGTGGGGGCCGTAATGAACCCCACTGAGGTCTTCTGCTCGGTCCCTGGA

```

## OUTPUT:

```
DNA data
      #T-A#
      #A---T#
      #A-----G#
      #T-----A#
      #A-----T#
      #A-----C#
      #C---A#
      #T-T#
      #T-C#
      #A---T#
      #C-----C#
      #A-----T#
      #C-----C#
      #A-----T#
      #G---A#
      #C-G#
      #C-C#
      #T---C#
      #A-----A#
      #A-----A#
      #A-----G#
      #G-----C#
      #G---A#
      #A---C#
```

---

## DISCUSSION AND FUTURE WORK:

### DISCUSSION:

- We have completed the project of “DNA visualization using Python.
- This project will show DNA double helix structure of a random person if his/her name is given as input.
- This project has input limitations.
- This program will take more time to run if we work with a large database of samples.

**FUTURE WORK:**

- DNA sequencing can be used in medical science for various purposes.
- Scientists can detect genome sequences.
- DNA sequencing can be used to detect the genes which are associated with some acquired diseases like cancers can be detected by observing certain genes.
- DNA sequencing is used for a range of purposes, including diagnosis and treatment of diseases. In general, sequencing allows health care practitioners to determine if a gene or the region that regulates a gene contains changes, called variants or mutations, that are linked to a disorder.

**CONCLUSION:**

We have successfully completed the project and are able to show the DNA double helix structure as output.

## APPENDIX:

## GITHUB LINK:

<https://github.com/jahid-cuet/DNA/blob/master/name.txt>

```
import pandas as pd
file = pd.read_csv('C:\\Users\\DELL\\Desktop\\country_2.txt',delimiter=' ')
file.head()
input_value=input()
result=file[file.Patient_Name == input_value]
if(len(result)):
    dna=result['dna_data'].to_string(index=False)
    print(dna)
else:
    print('Invalid')
file.loc[0]['dna_data']
Data = list(file['dna_data'])
print(Data)
dna2 = dna[ :-1]
print(len(dna2))
import random, sys, time
PAUSE = 0.15
ROWS = [

    '    ##, # Index 0 has no {}.',
    '    #{}-{}#',
    '    #{}---{}#',
    '    #{}-----{}#',
    '    #{}-----{}#',
    '    #{}-----{}#',
```

```

'  #{}-----{}#',
'  #{}---{}#',
'    #{}-{}#',
'      ##',    #9
'    #{}-{}#',
'  #{}---{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'    #{}---{}#',
'    #{}-{}#',
'      ##',    #18
'    #{}-{}#',
'  #{}---{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}---{}#',
'    #{}-{}#',
'      ##',    #27
'    #{}-{}#',
'  #{}---{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'  #{}-----{}#',
'    #{}---{}#',
'    #{}-{}#',
'      ##',    #36

```

```

'      #{}-{}#',
'      #{}---{}#',
'      #{}-----{}#',
'      #{}-----{}#',
'      #{}-----{}#',
'      #{}-----{}#',
'      #{}---{}#',
'      #{}-{}#',
'      ##', #45
'      #{}-{}#',
'      #{}---{}#',
'      #{}-----{}#',
'      #{}-----{}#',
'      #{}-----{}#',
'      #{}-----{}#',
'      #{}---{}#',
'      #{}-{}#',
'      ##',
]
print('DNA Animation')
print('Press Ctrl-C to quit.')
rowIndex = 0
for i,j in zip(dna,dna2):
    rowIndex = rowIndex + 1
    if rowIndex == len(ROWS):
        break
    # Row indexes 0 and 9 don't have nucleotides
    if rowIndex == 0 or rowIndex == 9 or rowIndex==18 or rowIndex==27 or rowIndex==36
or rowIndex==45:
        print(ROWS[rowIndex].format(i,j))

```



-

