



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DATA STRUCTURES AND ALGORITHMS (CSE 2003)

Winter Semester 2019-20

Faculty: Prof. Dr.Parveen Sultana H

TITLE: COMPARISON OF DNA

Nikhitha Perapola – 19BDS0125

B. Sai Sriya - 19BCE2350

Atluri Bhumika – 19BDS0109

Mutyam Sai Swithika – 19BCE2334

AIM

The main objective of this project is to perform two tasks:

- 1) It takes in DNA samples and presents the percentage of similarity between them by comparing them.
- 2) It generates the percentage of presence of a DNA characteristic in a particular DNA sample.

ABSTRACT

This project deals with DNA comparison - it inputs DNA sample's base data and compares them and displays the percentage of similarity between them. It also can calculate the percentage of presence of a particular DNA characteristic. This project carries a medical background usefulness and was primarily aimed to be launched in that field of use. It will hold paramount significance for the genetics department and serves to be a straight forward real world application. The details of the bases of the DNA samples must be required to perform the tasks or operations it offers. This can also contribute to the BioTechnology department or bio informatics under embedded technology.

DATA STRUCTURE USED:

Knuth Morris Pratt – KMP algorithm, it uses a degenerating property which improves the time complexity.

BASIC PRINCIPLE AND METHADODOLOGY

The two main types of nucleic acids are DNA and RNA.

DNA : Deoxyribonucleic acid is a thread-like chain of nucleotides carrying the genetic instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses.

Bases: The rules of base pairing (or nucleotide pairing) are: • A with T: the purine adenine (A) always pairs with the pyrimidine thymine (T) • C with G: the pyrimidine cytosine (C) always pairs with the purine guanine (G).

DNA Comparison: The project inputs DNA sample's base data and compares them and presents the percentage of similarity between them.

DNA Analysis: The project can calculate the percentage of presence of a particular DNA characteristic. The KMP algorithm is used to match the characteristics with the DNA sample and then the percentage is presented.

THE METHODS WHICH ARE USED AND EXECUTED IN WRITING THIS CODE ARE:

A) KMP Algorithm:

The Naive pattern searching algorithm doesn't work well in cases where we see many matching characters followed by a mismatching character. Following are some examples.

```
txt[] = "AAAAAAAAAAAAAAAAAAB"
```

```
pat[] = "AAAAB" txt[] = "ABABABCABABABCABABABC" pat[] = "ABABAC"
```

(not a worst case, but a bad case for Naive)

The KMP matching algorithm uses degenerating property (pattern having same sub patterns appearing more than once in the pattern) of the pattern and improves the worst case complexity to $O(n)$. The basic idea behind KMP's algorithm is: whenever we detect a mismatch (after some matches), we already know some of the characters in the text of next window. We take advantage of this information to avoid matching the characters that we know will anyway match. Input: txt[] = "THIS IS A TEST TEXT" pat[] = "TEST" Output: Pattern found at index 10

```
txt[] = "AABAACAADAABAABA"
```

```
pat[] = "AABA"
```

Output:

Pattern found at index

Pattern found at index 9

Pattern found at index 12

Text : A A B A A C A A D A A B A A B A

Pattern : A A B A

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| A | A | B | A | | | | | | A | A | B | A | | | |
| A | A | B | A | A | C | A | A | D | A | A | B | A | A | B | A |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | | | | | | | | | | | A | A | B | A | |

Pattern Found at 0, 9 and 12

In the project the KMP algorithm is used in determining the percentage of characteristics of a particular user inputted character.

B) Array based Stack and its functions for input and DNA comparison.

PROCESS MODULE

- 1. To check the percent matching of 2 DNA of same or different species, the first operation is to be performed.**

DNA is a double helix structure. It has two strands. We first input the number of nodes for the first (sample) DNA. Then the first strand is entered. Then the contents of the second strand is entered.

Note: Adenine(A) must always be paired with Thyamine(T) and Cytosine(C) is always paired with Guanine(G)

Then the procedure is repeated for the other DNA for which the percentage matching has to be checked.

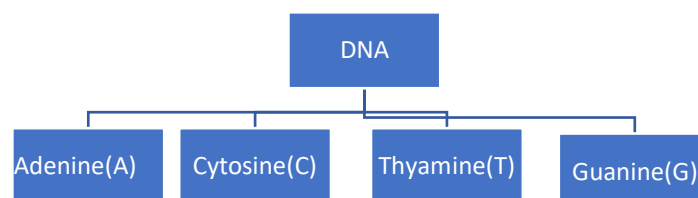
We get the desired output as the percentage similarity between the two DNA's.

- 2. The second operation is To check the percentage of characterstics present in the particular DNA.**

The input is entered in the same manner as in operation 1.

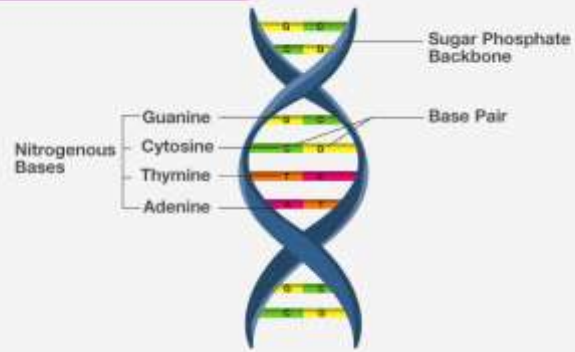
We then enter the length characteristics to be compared with the given sample. Next, the characters to be compared are entered.

It shows the desired percentage of the characteristics present in the given sample.



DNA STRUCTURE

BYJU'S
Learning App



EXPERIMENTAL CODE

```
#include<stdio.h>
#include<iostream>
#include<stdlib.h>
#include<conio.h>
#include<bits/stdc++.h>
using namespace std;
int main_page();
int about_page();
int welcome_page();
int instruct();
void computeLPSArray(char pat[], int M, int lps[]);
void KMPSearch(char* pat, char* txt, int M, int N);
void run_page();
void input();
void input1();
void comparing(char D1S1[], int count1, char D2S1[], int count2);
int inputcheck(char D1S1[], char D1S2[], int count); int
welcome_page()
{
system("cls");
cout<<"\n\n\n\n"<<"\t\t\t\t\t " <<"**WELCOME TO**\n"<<"\t\t\t\t\t"<<" DNA
COMPARISON ";
getch();
main_page();
}
int about_page()
{
system("cls");
cout<<"\n"<<"\t\t\t\t\t"<<"** DNA COMPARISON **";
cout<<"\nABOUT PAGE";
cout<<"\nThis is a page where you can perform two tasks"<<endl;
```

```

cout<<"1) To check the percentage matching or percentage of similarity of two DNA samples of
the same or different species"<<endl;
cout<<"2) To check the percentage of characteristics present in the particular DNA
sample\n\n\n\n";
cout<<endl<<"1.Main page"<<endl<<"2.Exit";
cout<<"\n Choose an option to proceed: "; int n;
cin>>n;
return n; }
int instruct()
{
system("cls");
cout<<" \n " <<"\t\t\t\t"<<"** DNA COMPARISON **"; cout<<"\nINSTRUCTIONS
PAGE\n";
cout<<"\n INPUT:\n"<<"\n\nThe DNA must be entered base by base, each in capitals. They are
stored as character array, not as string, so user must enter one base at a time.\n";
cout<<"\n OUTPUT: \n"<<"\n\nIn the comparing function, we compare the two inputted DNA
samples and the percentage of the DNA matched is displayed."<<"\n\nFor example, if the output
says 25% MATCH, it means 5% of DNA 1 is present in DNA 2. Hence, the species 2 will have
25% characteristics of species 1.";
cout<<"\n\nIn the matching characteristics function, we find the % of the input
character."<<"\n\nFor example if we say that there is 20% matching of the characteristics of the
data, then we can say that the species has 20% of that characteristic.";
cout<<endl<<"\n\n\n\n\n\n\n\n1.Main page"<<endl<<"2.Exit";
cout<<"\nChoose an option to proceed: ";
int n;
cin>>n;
return n;
}
void KMPSearch(char* pat, char* txt, int M, int N)
{
system("cls");
int counter=0;
int lps[M];
cout<<" \n " <<"\t\t\t\t"<<"** Device and Analyze **\n\n";
cout<<"\n\nResults of the characteristic comparison of the DNA sample: ";
computeLPSArray(pat, M, lps);

```

```

int i = 0;
int j = 0;
while(i < N)
{
if (pat[j] == txt[i])
{
    j++;
    i++;
}
if(j == M)
{
    counter++;
    j = lps[j-1];
}
else if(i < N && pat[j] != txt[i])
{
    if (j != 0)
        j = lps[j-1];
    else i = i+1;
}
}
if(counter==0)
cout<<"\n\nTHE GIVEN CHARACTERISTIC COULD NOT BE FOUND"<<endl;
else
{
double per=(counter*M*100)/N;
cout<<"\nTHE DNA HAD "<<per<<"% CHARACTERISTICS"<<endl;
}
}

void computeLPSArray(char pat[], int M, int lps[])
{
int len = 0;
lps[0] = 0;
int i = 1;

```



```

while (i < M)
{
if (pat[i] == pat[len])
{
    len++;
    lps[i] = len;
    i++;
}
else
{
    if (len != 0)
    {
        len = lps[len-1];
    }
    else
    {
        lps[i] = 0;
        i++;
    }
}
}
}

void run_page()
{
system("cls");
cout<<" \n "<<"\t\t\t\t"<<"** DNA COMPARISON **";
cout<<"\n OPERATIONS OFFERED";
int ch;
cout<<"\n1) To compare two DNAs and know how similar they are";
cout<<"\n2) To check the percentage of a given characteristics present in a DNA";
cout<<"\nChoose an option to proceed:";
cin>>ch;
switch(ch)
{

```

```

case 1: input1();
        break;

case 2: input();
        break;

default: cout<<"\n Invalid Input";
}
}

int main_page()
{
system("cls");
int n;
cout<<" \n "<<"\t\t\t\t"<<"** DNA COMPARISON **";
cout<<"\n1.About Us"<<endl<<"2.Instructions"<<endl<<"3.Run"<<endl<<"4.Exit"<<endl;
cout<<"Choose an option to proceed:";
cin>>n;
switch(n)
{
case 1: int a1;
        a1=about_page();
        if(a1==1)
        {
            main_page();
        }
        else if(a1==2)
        {
            return 0;
        }
        else
        {
            cout<<"invalid input";
        }
        break;

```

case 2:

```
    int a2;
    a2=instruct();
    if(a2==1)
        main_page();
    else if(a2==2)
        return 0;
    else
        cout<<"Invalid input";
    break;
```

case 3: run_page();

```
    break;
```

case 4: return 0;

```
default: cout<<"Invalid input\n" ;
```

```
}
```

```
return 0;
```

```
}
```

```
void input()
```

```
{
```

```
    system("cls");
```

```
    int i, m, count1=0, ch, len;
```

```
    char S1[80], S2[80], txt[35];
```

```
    cout<<" \n "<<"\t\t\t\t"<<"** DNA COMPARISON**"; cout<<"\nINPUT PAGE";
```

```
    cout<<"\nEnter the number of nodes in the DNA: ";
```

```
    cin>>m;
```

```
    count1=m*10;
```

```
    cout<<"\nEnter the first strand of the DNA: \n";
```

```
    for(i=0;i<count1;i++)
```

```
    {
```

```
        cin>>S1[i];
```

```
    }
```

```
    cout<<"\nEnter the second strand of the DNA: \n";
```

```

for(i=0;i<count1;i++)
{
    cin>>S2[i];
}
ch=inputcheck(S1,S2, count1);
if(ch==1)
{
    cout<<"\nThe first strand of the DNA: \n ";
    for(i=0;i<count1;i++)
    {
        cout<<S1[i]<<" ";
    }
    cout<<"\nThe second strand of the DNA:\n ";
    for(i=0;i<count1;i++)
    {
        cout<<S2[i]<<" ";
    }
}
if(ch!=1)
{
    cout<<"\nThe inputted DNA is incorrect. Kindly re-input.";
    input();
} else
{
    cout<<"\nEnter the length of the characteristics to be compared with the DNA sample";
    cin>>len;
    cout<<"\nEnter the characteristics to be compared with the DNA sample";
    for(i=0;i<len;i++)
    {
        cin>>txt[i];
    }
    getch();
    KMPSearch(S1,txt, len, count1);
}

```

```

}
void input1()
{
system("cls");
int i, m1, m2, count1=0, count2=0, ch1, ch2;
char D1S1[80], D1S2[80], D2S1[80], D2S2[80];
cout<<" \n "<<"\t\t\t\t"<<"** DNA COMPARISON **"; cout<<"\nINPUT PAGE";
cout<<"\nEnter the number of nodes in the DNA 1: ";
cin>>m1;
count1=m1*10;
cout<<"\nEnter the first strand of the DNA 1: \n";
for(i=0;i<count1;i++)
{
    cin>>D1S1[i];
}
cout<<"\nEnter the second strand of the DNA 1: \n";
for(i=0;i<count1;i++)
{
    cin>>D1S2[i];
}
ch1=inputcheck(D1S1,D1S2, count1);
cout<<"\nEnter the number of nodes in the DNA 2: ";
cin>>m2;
cout<<"\nEnter the first strand of the DNA 2: \n";
count2=m2*10;
for(i=0;i<count2;i++)
{
    cin>>D2S1[i];
}
cout<<"\nEnter the second strand of the DNA 2: \n";
for(i=0;i<count2;i++)
{
    cin>>D2S2[i];
}

```

```

ch2=inputcheck(D2S1,D2S2, count2);
if(ch1==1)
{
    cout<<"\nThe first strand of the DNA 1 : \n ";
    for(i=0;i<count1;i++)
    {
        cout<<D1S1[i]<<" ";
    }
    cout<<"\nThe second strand of the DNA 1:\n ";
    for(i=0;i<count1;i++)
    {
        cout<<D1S2[i]<<" ";
    }
}
if(ch2==1)
{
    cout<<"\nThe first strand of the DNA 2 : \n ";
    for(i=0;i<count2;i++)
    {
        cout<<D2S1[i]<<" ";
    }
    cout<<"\nThe second strand of the DNA 2:\n ";
    for(i=0;i<count2;i++)
    {
        cout<<D2S2[i]<<" ";
    }
}
if(ch1!=1 || ch2!=1)
{
    cout<<"\n The inputted DNA is incorrect. Kindly re-input.";
    input1();
}
getch();
comparing(D1S1, count1, D2S1, count2);

```

```

}
void comparing(char D1S1[], int count1, char D2S1[], int count2)
{
system("cls");
int i, a=0,b=0, m;
cout<<" \n "<<"\t\t\t\t"<<"** DNA COMPARISON **\n\n";
cout<<"\nRESULTS OF DNA SAMPLE COMPARISON";
if(count1!=count2)
{
    cout<<"\nThe two DNA's cannot be compared as the strand lengths vary.";
}
else
{
    for(i=0; i<count1;i++)
    {
        if(D1S1[i]==D2S1[i])
        {
            a++;
        }
        else
        {
            b++;
        }
    }
    m=((a)/(a+b))*100;
    cout<<"\n\n \nFrom the comparison of the two DNA samples, it is identified that the percentage
of similarity between the two DNAs is "<<m;
}
}
int inputcheck(char D1S1[], char D1S2[], int count)
{
int i, a=0, b=0, m=0;
for(i=0;i<count;i++)
{

```

```
if(D1S1[i]=='A')
{
    if(D1S2[i]=='T')
    {
        a++;
    }
    else
    {
        b++;
        break;
    }
}
else
if(D1S1[i]=='T')
{
    if(D1S2[i]=='A')
    {
        a++;
    }
    else
    {
        b++;
        break;
    }
}
else if(D1S1[i]=='G')
{
    if(D1S2[i]=='C')
    {
        a++;
    }
    else
    {
        b++;
    }
}
```



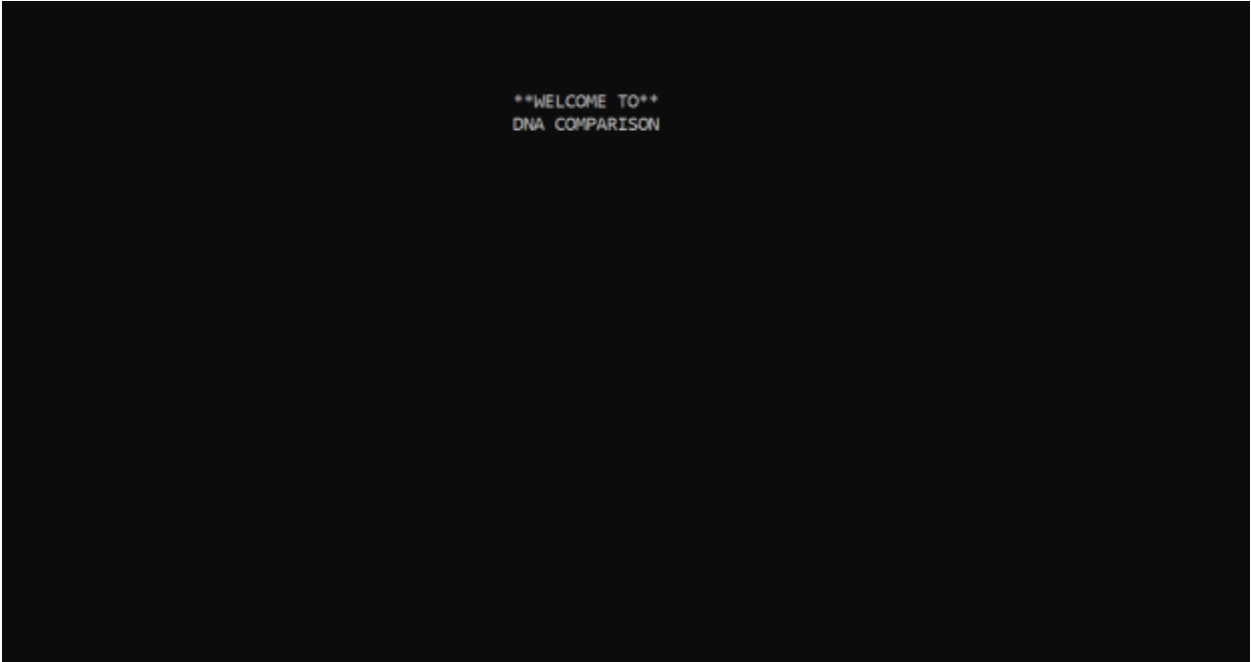
```

        break;
    }
}
else if(D1S1[i]=='C')
{
    if(D1S2[i]=='G')
    {
        a++;
    }
    else
    {
        b++;
        break;
    }
}
else
{
    m++;
}
}
if(b==0 && m==0)
{
    cout<<"\n The Inputted DNA is correct";
    return 1;
}
else
{
    cout<<"\n The Inputted DNA is incorrect";
    return 0;
}
}
int main()
{
    welcome_page();

```

```
getch();  
system("cls");  
return 0;  
}
```

OUTPUT



```
**WELCOME TO**  
DNA COMPARISON
```

**** DNA COMPARISON ****

1.About Us
2.Instructions
3.Run
4.Exit
Choose an option to proceed:

**** DNA COMPARISON ****

ABOUT PAGE
This is a page where you can perform two tasks
1) To check the percentage matching or percentage of similarity of two DNA samples of the same or different species
2) To check the percentage of characteristics present in the particular DNA sample

1.Main page
2.Exit
Choose an option to proceed:

```

                                ** DNA COMPARISON **

INSTRUCTIONS PAGE

INPUT:

The DNA must be entered base by base, each in capitals. They are stored as character array, not as string, so user must
enter one base at a time.

OUTPUT:

In the comparing function, we compare the two inputted DNA samples and the percentage of the DNA matched is displayed.
For example, if the output says 25% MATCH, it means 5% of DNA 1 is present in DNA 2. Hence, the species 2 will have 25%
characteristics of species 1.
In the matching characteristics function, we find the % of the input character.
For example if we say that there is 20% matching of the characteristics of the data, then we can say that the species ha
s 20% of that characteristic.


1.Main page
2.Exit
Choose an option to proceed:1

```

```

                                ** DNA COMPARISON **

OPERATIONS OFFERED
1) To compare two DNAs and know how similar they are
2) To check the percentage of a given characteristics present in a DNA
Choose an option to proceed:

```

For percentage of similarity between DNA samples by comparison:

i)

```

** DNA COMPARISON **

INPUT PAGE
Enter the number of nodes in the DNA 1: 1

Enter the first strand of the DNA 1:
A
A
A
A
A
T
T
T
T

Enter the second strand of the DNA 1:
T
T
T
T
A
A
A
A
A

The Inputted DNA is correct
```

```

A

The Inputted DNA is correct
Enter the number of nodes in the DNA 2: 1

Enter the first strand of the DNA 2:
C
G
C
G
C
G
C
G
C
G

Enter the second strand of the DNA 2:
G
C
G
C
G
C
G
C
G
C

The Inputted DNA is correct
```

```
C
G
Enter the second strand of the DNA 2:
G
C
G
C
G
C
G
C
G
C
C
```

```
The Inputted DNA is correct
The first strand of the DNA 1 :
A A A A A T T T T T
The second strand of the DNA 1:
T T T T T A A A A A
The first strand of the DNA 2 :
C G C G C G C G C G
The second strand of the DNA 2:
G C G C G C G C G C
```

```
** DNA COMPARISON **
```

```
RESULTS OF DNA SAMPLE COMPARISON
```

```
From the comparison of the two DNA samples, it is identified that the percentage of similarity between the two DNAs is 80%
```

ii)

```
                                ** DNA COMPARISON **
INPUT PAGE
Enter the number of nodes in the DNA 1: 2

Enter the first strand of the DNA 1:
A
T
G
G
T
G
C
C
T
C
T
G
A
C
T
C
C
T
G
A

Enter the second strand of the DNA 1:
```

```
C
C
T
G
A

Enter the second strand of the DNA 1:
T
A
C
C
A
C
G
G
A
G
A
C
T
G
A
G
G
A
C
T

The Inputted DNA is correct
Enter the number of nodes in the DNA 2:
```

```
G
G
A
C
T

The Inputted DNA is correct
Enter the number of nodes in the DNA 2: 2

Enter the first strand of the DNA 2:
A
T
G
G
T
C
C
T
C
T
G
A
C
C
T
G
A
```

```
T
A
C
C
A
C
G
G
A
G
A
G
A
C
T
G
A
G
G
A
C
T

The Inputted DNA is correct
The first strand of the DNA 1 :
A T G G T G C C T C T G A C T C C T G A
The second strand of the DNA 1:
T A C C A C G G A G A C T G A G G A C T
The first strand of the DNA 2 :
A T G G T G C C T C T G A C T C C T G A
The second strand of the DNA 2:
T A C C A C G G A G A C T G A G G A C T
```



```

** DNA COMPARISON **

RESULTS OF DNA SAMPLE COMPARISON

From the comparison of the two DNA samples, it is identified that the percentage of similarity between the two DNAs is 100

```

For percentage of characteristics present in the DNA samples:

i)

```

** DNA COMPARISON**

INPUT PAGE
Enter the number of nodes in the DNA: 1

Enter the first strand of the DNA:
A
A
A
A
A
T
T
T
T
T

Enter the second strand of the DNA:
T
T
T
T
T
A
A
A
A
A

The Inputted DNA is correct

```

A
T
T
T
T
T
T

Enter the second strand of the DNA:

T
T
T
T
T
A
A
A
A
A

The Inputted DNA is correct

The first strand of the DNA:

A A A A A T T T T T

The second strand of the DNA:

T T T T T A A A A A

Enter the length of the characteristics to be compared with the DNA sample3

Enter the characteristics to be compared with the DNA sampleA

A

A

** Device and Analyze **

Results of the characteristic comparison of the DNA sample:

THE DNA HAD 38% CHARACTERISTICS

ii)

```

                                                    ** DNA COMPARISON**
INPUT PAGE
Enter the number of nodes in the DNA: 1

Enter the first strand of the DNA:
A
A
A
A
A
A
A
T
T
T
T

Enter the second strand of the DNA:
T
T
T
T
T
T
T
A
A
A
A

The Inputted DNA is correct

Enter the second strand of the DNA:
T
T
T
T
T
T
T
A
A
A
A

The Inputted DNA is correct
The first strand of the DNA:
A A A A A T T T T
The second strand of the DNA:
T T T T T T A A A A
Enter the length of the characteristics to be compared with the DNA sample5

Enter the characteristics to be compared with the DNA sampleA
A
A
A
A

```

```

                                ** Device and Analyze **

Results of the characteristic comparison of the DNA sample:
THE DNA HAD 50% CHARACTERISTICS

```

CONCLUSION

From the above working model, we infer two results:

The exact percentage of similarity between the given two DNA samples and the percentage of presence of a particular DNA characteristic in the inputted DNA sample. This project carries a medical background usefulness and was primarily aimed to be launched in that field of use only.

REFERENCE MATERIAL

1. C++ by Sumita Arora
2. Geeks for Geeks <https://ide.geeksforgeeks.org/>
3. <https://gist.github.com/cagdass/ede868d39b5c18485a7a094a03b1931c>
4. Using C++ by Bjarne Stroustrup
5. KMP Algorithm: <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>

6. <https://www.ics.uci.edu/~eppstein/161/960227.html>
7. DNA matching algorithms:
8. <https://www.sciencedirect.com/science/article/pii/S0957417417301811>
9. Comparison of Three pattern matching algorithms:
10. <http://ijsetr.com/uploads/625413IJSETR2868-162.pdf>