K L UNIVERSITY

FRESHMAN ENGINEERING DEPARTMENT

A Project Based Lab Report

On

FIBONACCI STRINGS

SUBMITTED BY:

I.D NUMBER	NAME
2100090187	Niranjan sah
2100090186	Snigdha sindusha mantha
2100050043	Siraparapu Yogendra
2100050065	Pillala Sri ram

UNDER THE ESTEEMED GUIDANCE OF

Mrs.Yamini

ASST.PROFESSOR

KL UNIVERSITY

Green fields, Vaddeswaram – 522 502
Guntur Dt., AP, India.

DEPARTMENT OF BASIC ENGINEERING SCIENCES

CERTIFICATE

This is to certify that the project based laboratory report entitle“FIBONACCI STRINGS” submitted by Mr. **Niranjan, Mrs. Snigdha, Mr. Yogendra, Mr.Sri ram** bearing Regd.No.2100090187,2100090186,2100050043,2100050065 to the **Department of Basic Engineering Sciences, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in “Technical Skills-I(Coding)” course in I B Tech I Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2021 – 2022.

PROJECT SUPERVISOR DEPARTMENT	HEAD OF THE DEPARTMENT
Mrs.Yamini	Dr. D.HARITHA

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our Director **Dr. A. Jagadeesh** for his administration towards our academic growth.

I express sincere gratitude to our Coordinator and HOD-BES **Dr. D.Haritha** for her leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Yamini** for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

I.D NUMBER	NAME
2100090187	Niranjan sah
2100090186	Snigdha sindusha mantha
2100050043	Siraparapu Yogendra
2100050065	Pillala Sri ram

ABSTRACT

Fibonacci strings are defined as follows: f1 = «a» f2 = «b» fn = fn - 1 fn - 2, n > 2 Thus, the first five Fibonacci strings are: "a", "b", "ba", "bab", "babba". You are given a Fibonacci string and m strings si. For each string si, find the number of times it occurs in the given Fibonacci string as a substring.

Input Format:

The first line contains two space-separated integers k and m — the

number of a Fibonacci string and the number of queries, correspondingly. Next m lines contain strings si that correspond to the queries. It is guaranteed that strings si aren't empty and consist only of characters "a" and "b". The input limitations for getting 30 points are: 1 ≤ k ≤ 3000 1 ≤ m ≤ 3000 The total length of strings si doesn't exceed 3000 The input limitations for getting 100 points are: 1 ≤ k ≤ 1018 1 ≤ m ≤ 104 The total length of strings si doesn't exceed 105 Please do not use the %lld specifier to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the %I64dspecifier.

Output Format:

For each string si print the number of times it occurs in the given Fibonacci

string as a substring. Since the numbers can be large enough, print them modulo 1000000007 (109 + 7). Print the answers for the strings in the order in which they are given in the input.

INDEX

INTRODUCTION

Fibonacci strings are defined as follows: f1 = «a» f2 = «b» fn = fn - 1 fn - 2, n > 2 Thus, the first five Fibonacci strings are: "a", "b", "ba", "bab", "babba". You are given a Fibonacci string and m strings si. For each string si, find the number of times it occurs in the given Fibonacci string as a substring.

AIM

It is defined by setting each term equal to sum of two previous term

Advantages:-

1. It’s Composed Of Integers

For estimating the time it takes to complete tasks, you want a scale that is made of integers. The goal of estimating tasks in Agile is a high-level estimate. There’s no need for the granularity of 2.4 and 3.2. Integers, or whole numbers, are all that is needed. Giving the option of half-steps, quarter steps, and so on, only slows your team down in minutia.

2. It’s Exponential

When estimating time, the shorter the time span the more certainty. Longer tasks are more complex and time estimates are less precise. An exponential (grows at an increasing rate) scale provides detail for small tasks and forces uncertainty for large tasks.

Furthermore, because the Fibonacci Sequence is a lagging exponential sequence, it provides detail for small tasks (1,2,3) which are simple to estimate, but starts to provide less detail for medium tasks (5,8), and even less detail for large tasks (13,21). This forces your team to have more detail when estimating small tasks (is it a 1 of a 2?) and less when faced with large tasks (hmmm, I wonder if that is a 13 or a 21…). This helps build your estimates with increasing uncertainty as time estimates get longer, creating a more efficient, and effective estimation.

3. It Forces You To Choose "More Or Less"

In addition to building in uncertainty for increased time spans, the Fibonacci sequence also forces your team to make a choice. When faced with a larger task, "is it a, 8, a 13 or a 21?", there is no in-between. This helps your team group and differentiate the size of tasks.

Another aspect of the Fibonacci sequence is the distance between points. 3 to 5 is a difference of 2, but 5 to 8 is a difference of 3. This allows your brain to intuitively distinguish between the numbers of the Fibonacci scale as different magnitudes.

4. It’s Non-Linear

Finally, the nonlinear nature of the Fibonacci scales reduces over-analysis. 4 out of the 6 numbers used are prime numbers, reducing your ability to evenly break down or compare tasks. Large tasks are not squarely related to one another (that’s twice as long as that), and the numbers don’t give the impression that if you just had multiple people work on it, the task would be twice as fast. This helps reduce over-analysis, or "analysis paralysis".

5. It Sounds Cool And Adds An Air Of Legitimacy

Ok, this isn’t exactly a concrete reason that the Fibonacci Sequence is a better scale to use than others, but the word "Fibonacci" does sound cool and can help with adoption, and buy-in. The sequence is simple and easy to remember, encouraging team adoption and use. Additionally, the prestigious and somewhat mysterious nature of the sequence adds an air of legitimacy to the method, making it easier to get team and executive buy-in.

Disadvantages:-

1.It is fixed size .

2.It is less intuitive notation for library feature.

3.Primitive c arrays do not track their own size

Future enhancements:-

i) Adding permanent memory to the system.

ii) For that compiler is going store all the data in memory.

iii) When we give any name to the compiler, if it is already stored in the memory it will correct our name according to the given instructions.

SYSTEM REQUIREMENTS

○ SOFTWARE REQUIREMENTS:

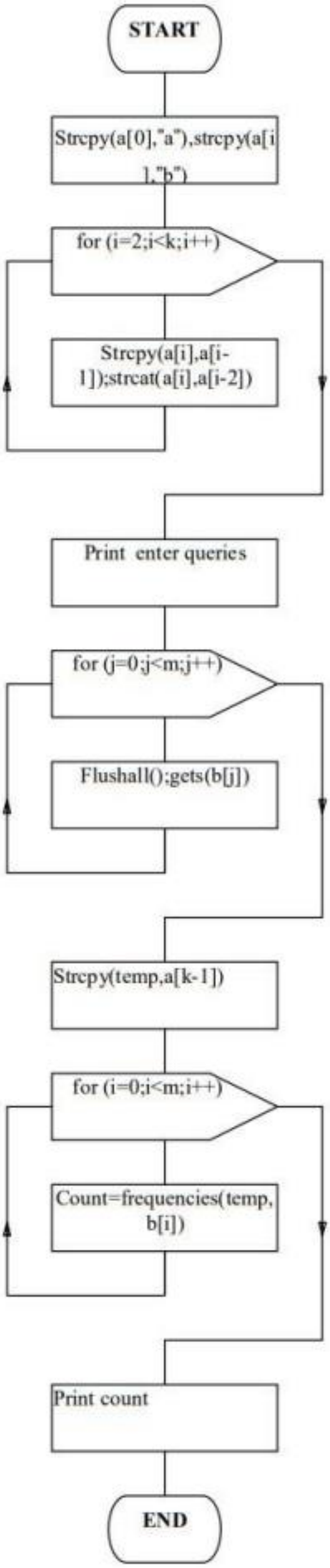
The major software requirements of the project are as follows:
Language : Turbo-C, dev c++, gdb online compiler etc;
Operating system**:** Windows Xp or later.

○ HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows:

Ram:512MB

Processor:Intelcore i


START
Strcpy(a[0],"a"),strcpy(a[i
],"b")
for (i=2;i<k;i++)
Strcpy(a[i],a[i-1]);strcat(a[i],a[i-2])
Print enter queries
for (j=0;j<m;j++)
Flushall();gets(b[j])
Strcpy(temp,a[k-1])
for (i=0;i<m;i++)
Count=frequencies(temp, b[i])
Print count
END

ALGORITHM

```
STEP1: Start
STEP2:  Read  Strcpy(a[0],”a”)
STEP3:   Read Strcpy(a[i],”b”)
STEP4:  k,m values
STEP5:  Int i=2
STEP6:  if i<k
            6(a):if it is true Go to step7
            6(b):if it is false Go to step8
STEP7: Strcpy(a[i],a[i-1])
            7(a):Strcat(a[i],a[i-2])
            7(b):Increment i++
            7(c):And again Go to step16
STEP8: Print quieries
            8(a):j=0
STEP9: if j<m
            9(a):if it is true Go to step10
            9(b):if it is false Go to step11
STEP10: gets b[j]
            10(a):Increment j++
            10(b):And again Go to step9
STEP11: Strcpy(temp,a[k-1])
              11(a):int i=0
STEP12: if i<m

12(a):if it is true Go to step13 12

(b):if it is false Go to step24
```

```
STEP13: count frequencies(temp,b[i])

        13(a):stringlen=strlen(str)
        13(b):searchlen=strlen(search)
STEP14: count=0
        14(a):int i=0
STEP15: if i<str_lenscaler
        15(a):if it is true Go to step16
        15(b):if it is true Go to step21
STEP16: found=1
        16(a):int i=0
STEP17: if j<searchlen
        17(a):if it is true Go to step18
        17(b):if it is true Go to step20
STEP18: Str[i+j]!=search[j]
        18(a):found=0
        18(b):break
STEP19: Increment j++
        19(a):And again Go to step17
STEP20: found==1
        20(a):count++
        20(b):Increment i++
        20(c):And again Go to step15
STEP21: return count
STEP22: Go to step13
STEP23: Print count
        23(a):Increment i++
        23(b):And again Go to step12
STEP24:Stop
```

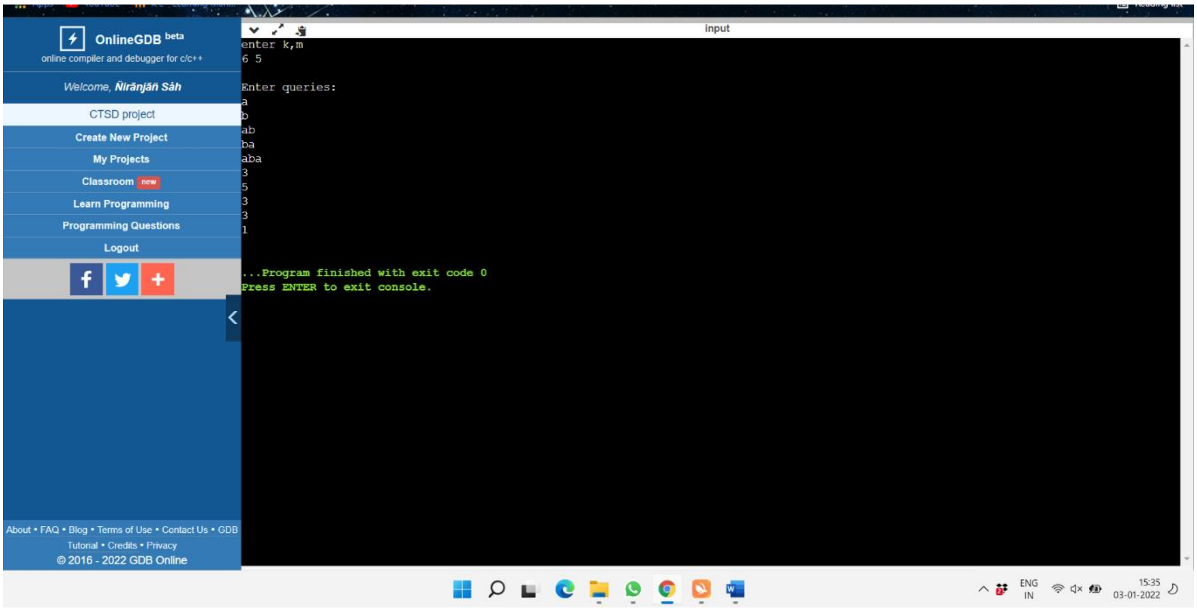
IMPLEMENTATION

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

int frequencies(char * str, char * Search);

int  main()

{
char a[1000][40],b[1000][40],temp[100];

int k,m,i,j,count;

strcpy(a[0],"a");

strcpy(a[1],"b");

printf("enter k,m\n");

scanf("%d%d",&k,&m);

for(i=2;i<k;i++)

{
strcpy(a[i],a[i-1]);

strcat(a[i],a[i-2]);

}

printf("\nenter queries\n");

for(j=0;j<m;j++)

{
scanf("%s",b[j]);

}
strcpy(temp,a[k-1]);

for(i=0;i<m;i++)

{
count = frequencies(temp, b[i]);

printf("%d\n",count);

}
return 0;
}
int frequencies(char * str, char * Search)
{
int i, j, found, count;
  int len, searchLen;
len = strlen(str);
searchLen = strlen(Search);
count = 0;
for(i=0; i<= len-searchLen; i++)
{
found = 1;
   for(j=0; j<searchLen; j++)
   {
   if(str[i + j] != Search[j])
{
found = 0;
   break;
}
}
if(found == 1)
  {
   count++;
  }
}
return count;

}
```

INTEGRATION AND SYSTEM TESTING

OUTPUTS

Screen Shots:



```
enter k,m
6 5

Enter queries:
a
b
ab
ba
aba
3
5
3
3
1


...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION

we have successfully completed the project "FIBONACCI STRINGS" in simple way. And given all test cases and everything is passed.