NAME : VISHESH CHOUHAN

ENROLLMENT NO : 0801CS211101

CLASS : B.Tech II YEAR

SUBJECT : PROGRAMMING PRACTICES

TOPIC : MINIPROJECT

PROJECT TITLE : STRING MANIPULATOR

Objectives of project

To create a library of functions for string manipulation

Function description

- **isupper** Returns true if the given character is a uppercase character.
- **islower** Returns true if the given character is a lower character.
- **isspace** Returns true if the given character is a space character.
- **upper** Returns the uppercase character, of the given character.
- **lower** Returns the lowercase character, of the given character.
- **len** Returns the length of the given string.
- **touppercase** Returns the uppercase string, of the given stirng.
- **tolowercase** Returns the uppercase string, of the given string.
- **capitalizecase** Returns the capitalized string, given a string.
- **sentencecase** Returns the string in sentence case, given a string.
- **togglecase** Returns the string whose case are toggled with respect to the initial string, given a string.
- **issame** Returns true if the given two strings are same, false otherwise.
- **reversestr** Returns a string that is in reverse order as the given string.
- **ispalindromic** Returns true if the input string is palindromic, false otherwise.
- **index** Returns the first occurrence of a character in a string, returns -1 if character not found.
- **indexstr** Returns the first occurrence of a string in a string, returns -1 if string not found.
- **main** It is the runner of the code. It initializes the program.

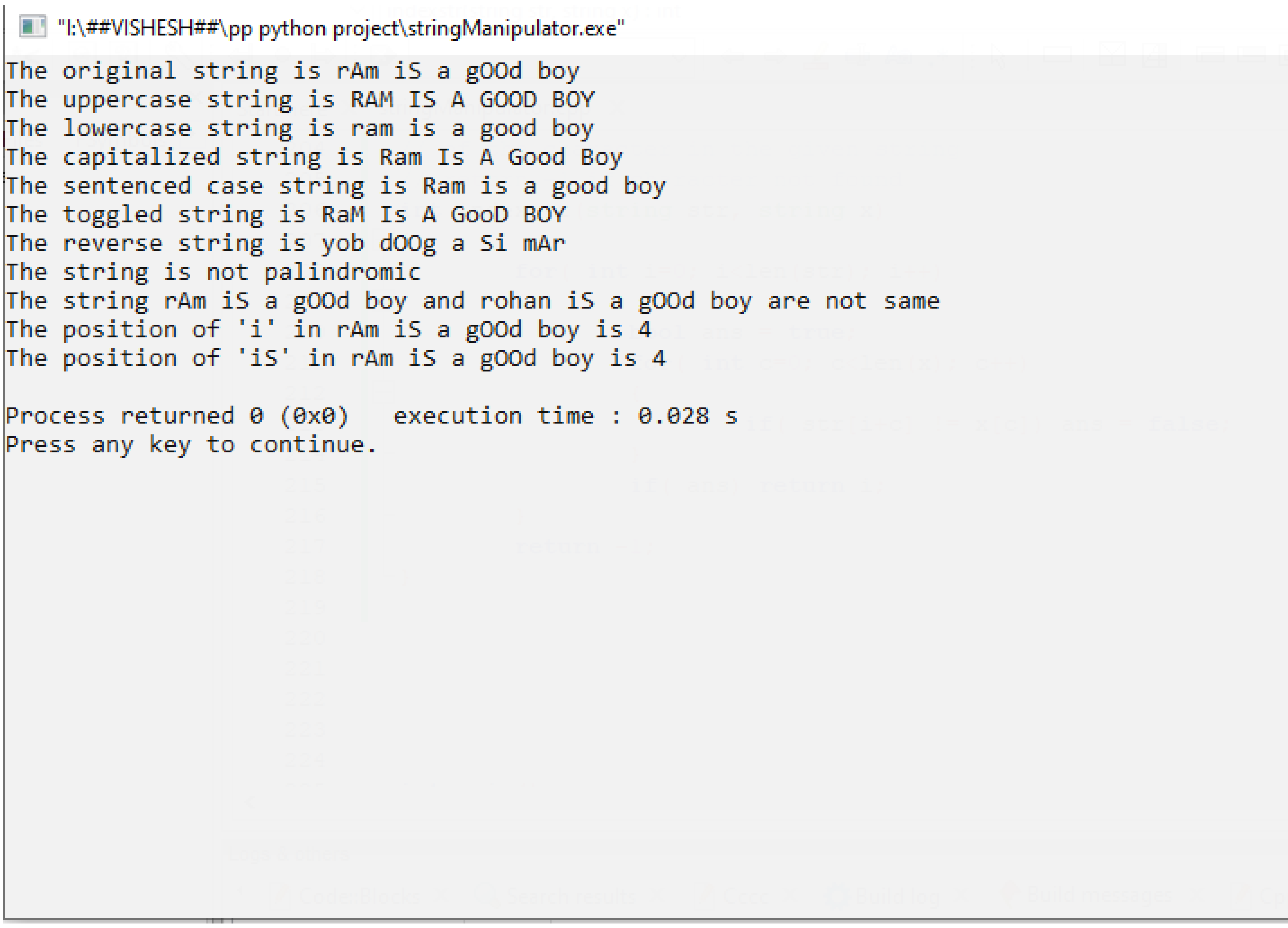
PROGRAM CODE

```
#include<iostream>
#include<stdlib.h>
#include <cstdlib>
using namespace std;

// function that checks the character
// is in upper case or not
bool isupper(char ch)
{
        if(ch>=65 && ch<=90)
        {
                return true;
        }
        return false;

}

// function that checks the character
// is in upper case or not
bool islower(char ch)
{
        if(ch>=90 && ch<=122)
        {
                return true;
        }
        return false;
}

// function that checks the character
// is a space or not
bool isspace(char ch)
{
        if(ch == 32)
        {
                return true;
        }
        return false;
}


// function that converts the character
// into upper case
char upper(char ch)
{
        if(islower(ch))
        {
                return ch-32;
        }
        return ch;
}

// function that converts the character
// into lower case
```

```
char lower(char ch)
{
        if(isupper(ch))
        {
                return ch+32;
        }
        return ch;
}



// function that finds the length of the string
int len(string str)
{
        int l=0;
        for(char x:str)
        {
                l++;
        }
        return l;
}


// converts the string into uppercase
string touppercase(string str)
{
        for(int i=0;i<len(str);i++)
        {
                if(islower(str[i]))
                {
                        str[i] = upper(str[i]);
                }
        }

        return str;
}


// converts the string into lowercase
string tolowercase(string str)
{
        for(int i=0;i<len(str);i++)
        {
                if(isupper(str[i]))
                {
                        str[i] = lower(str[i]);
                }
        }
        return str;
}

// Thsi function converts the string into
// "Capitalize each word" case
string capitalizecase(string str)
{
        for(int i=0;i<len(str)-1;i++)
        {
                if( isspace(str[i]) && islower(str[i+1]) )
                {
                        str[i+1] = upper(str[i+1]);
                }
                else
                {
                       str[i+1] = lower(str[i+1]);
                }
        }
        str[0] = upper(str[0]);
        return str;
}

// This function convert the string in to sentence case
string sentencecase(string str)
{
        for(int i=0;i<len(str);i++)
        {
                if(isupper(str[i]) )
                {
                        str[i] = lower(str[i]);
                }

        }
        str[0] = upper(str[0]);
        return str;
}


// This function toggle the case
// of the character in string
// and returns the string
string togglecase(string str)
{
        for(int i=0;i<len(str);i++)
        {
                if(isupper(str[i]) )
                {
                        str[i] = lower(str[i]);
                }
                else if(islower(str[i]) )
                {
                        str[i] = upper(str[i]);
                }

        }
        str[0] = upper(str[0]);
        return str;
}

// finds out whether the two strings are same or not
bool issame(string a, string b)
{
        if( len(a)!= len(b)) return false;
        for( int i=0; i< len(a); i++)
        {
                if( a[i] != b[i]) return false;
        }
        return true;
}

// reversestr() returns the string into reverse order
string reversestr(string str)
{
        string rev = str;
        for( int i=0; i<len(str); i++)
        {
                rev[i] = str[len(str)-1-i];
        }
        return rev;
}

// Checks whether the string is palindromic or not
bool ispalindromic(string str)
{
        if( issame( str, reversestr(str))) return true;
        else return false;
}


// index() returns the position of the first occurrence
// of the character in the given string
// return -1 if character not found
int index(string str, char x)
{
        for( int i=0; i<len(str); i++)
        {
                if( str[i] == x) return i;
        }
        return -1;
}


// index() returns the position of the first occurrence
// of the character in the given string
// return -1 if character not found
int indexstr(string str, string x)
{
        for( int i=0; i<len(str); i++)
        {
                bool ans = true;
                for( int c=0; c<len(x); c++)
                {
                        if( str[i+c] != x[c]) ans = false;
                }
                if( ans) return i;
        }
        return -1;
}
```

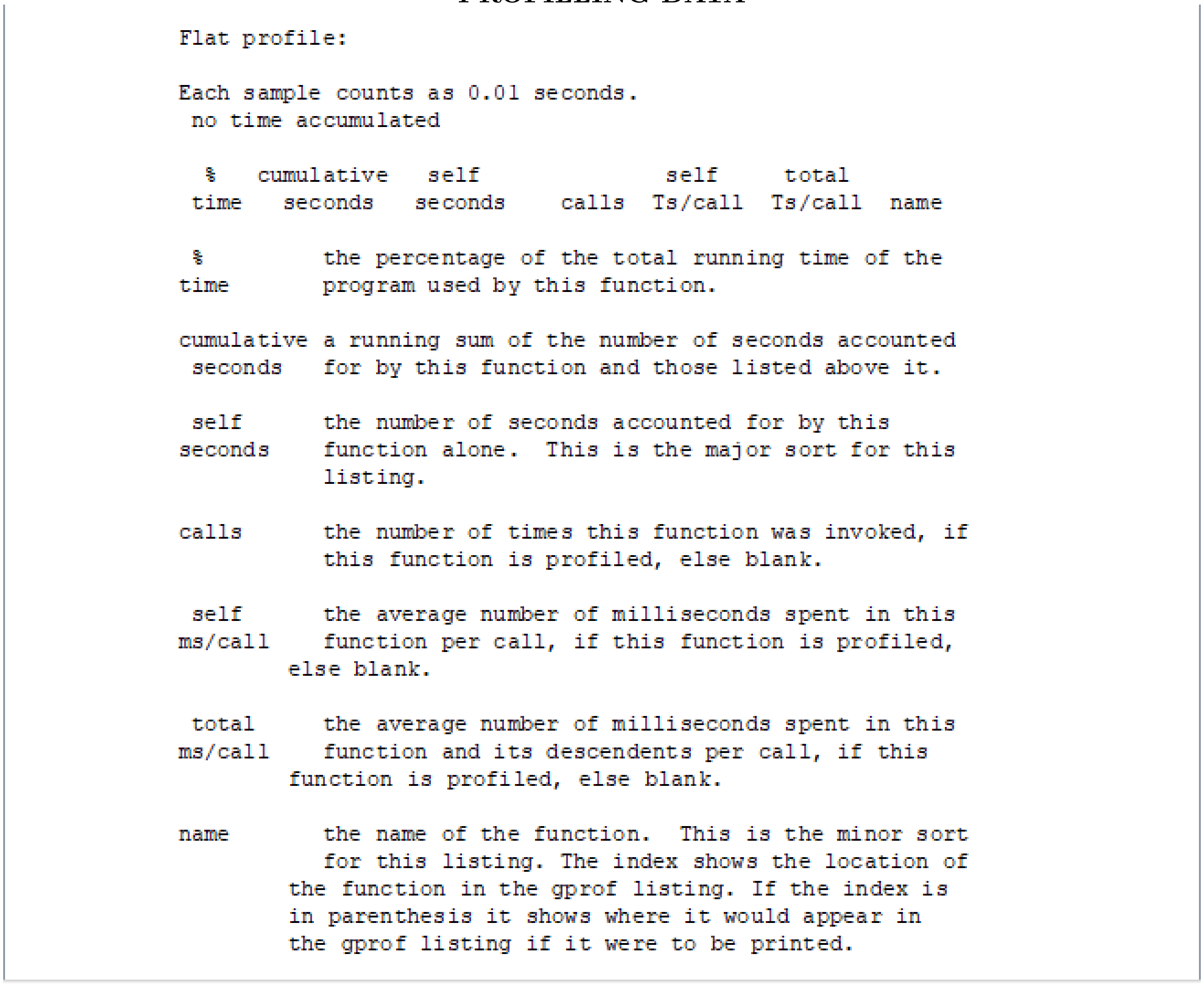
```
int main()
{
        string str = "rAm␣iS␣a␣gOOd␣boy";

        string str2 = "rohan␣iS␣a␣gOOd␣boy";

        cout<<"The␣original␣string␣is␣"<<str<<endl;


        cout<<"The␣uppercase␣string␣is␣"<<touppercase(str)<<endl;

        cout<<"The␣lowercase␣string␣is␣"<<tolowercase(str)<<endl;

        cout<<"The␣capitalized␣string␣is␣"<<capitalizecase(str)<<endl;

        cout<<"The␣sentenced␣case␣string␣is␣"<<sentencecase(str)<<endl;

        cout<<"The␣toggled␣string␣is␣"<<togglecase(str)<<endl;


        cout<<"The␣reverse␣string␣is␣"<<reversestr(str)<<endl;

        if(ispalindromic(str))
        {
                cout<<"The␣string␣is␣palindromic"<<endl;
        }
        else
        {
                cout<<"The␣string␣is␣not␣palindromic"<<endl;
        }

        if(issame(str,str2))
        {
                cout<<"The␣string␣"<<str<<"␣and␣"<< str2<<"␣are␣same"<<endl;
        }
        else
        {
                cout<<"The␣string␣"<<str<<"␣and␣"<< str2<<"␣are␣not␣same"<<endl;
        }

        cout<<"The␣position␣of␣'i'␣in␣"<<str<<"␣is␣"<<index(str,'i')<<endl;
        cout<<"The␣position␣of␣'iS'␣in␣"<<str<<"␣is␣"<<indexstr(str,"iS")<<endl;

        return 0;
        exit(0);
}
```

PROGRAM OUTPUT


"I:\##VISHESH##\pp python project\stringManipulator.exe"

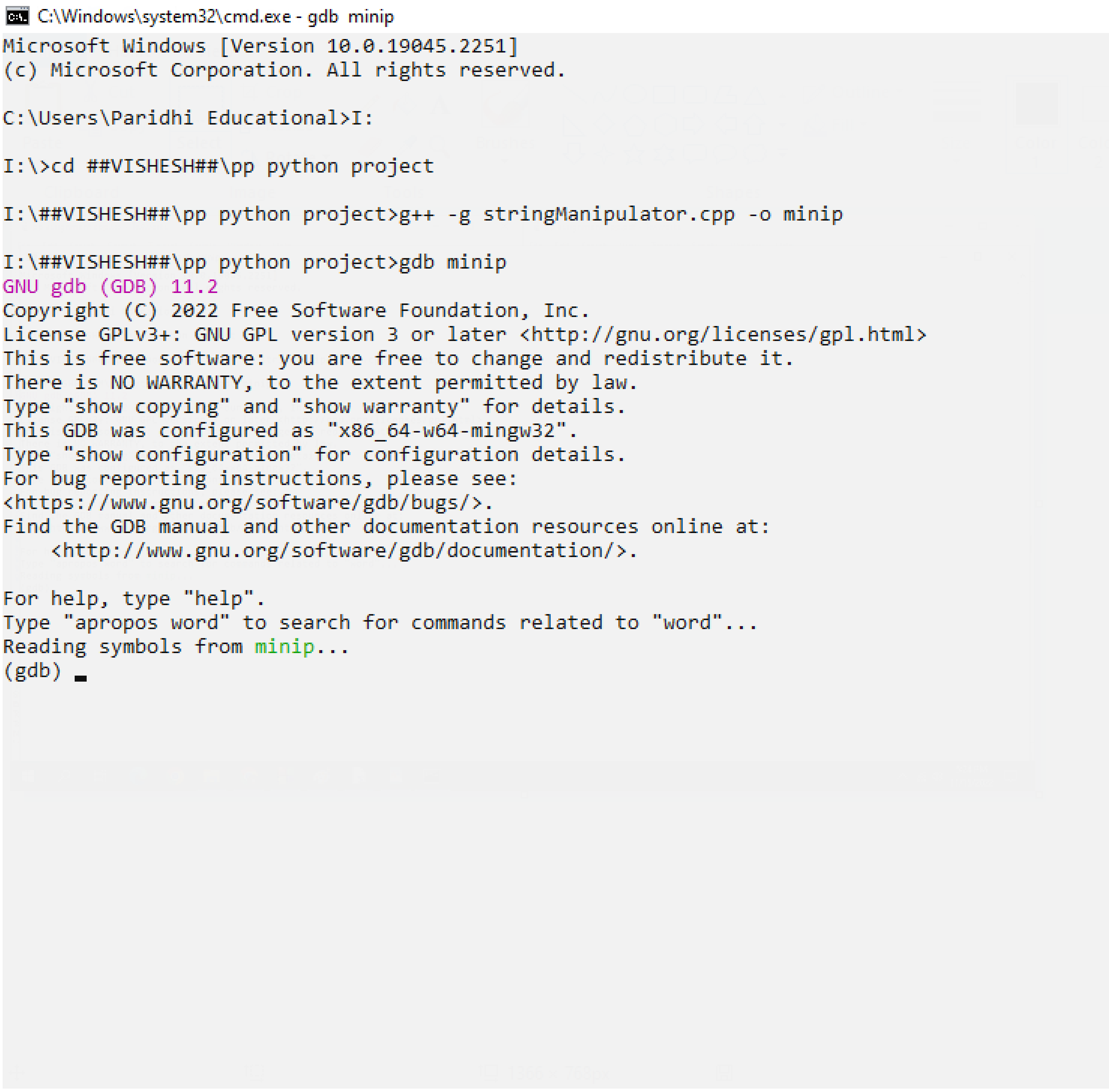
```
The original string is rAm iS a gOOd boy
The uppercase string is RAM IS A GOOD BOY
The lowercase string is ram is a good boy
The capitalized string is Ram Is A Good Boy
The sentenced case string is Ram is a good boy
The toggled string is RaM Is A GooD BOY
The reverse string is yob dOOg a Si mAr
The string is not palindromic
The string rAm iS a gOOd boy and rohan iS a gOOd boy are not same
The position of 'i' in rAm iS a gOOd boy is 4
The position of 'iS' in rAm iS a gOOd boy is 4

Process returned 0 (0x0)   execution time : 0.028 s
Press any key to continue.
```

PROFILLING DATA



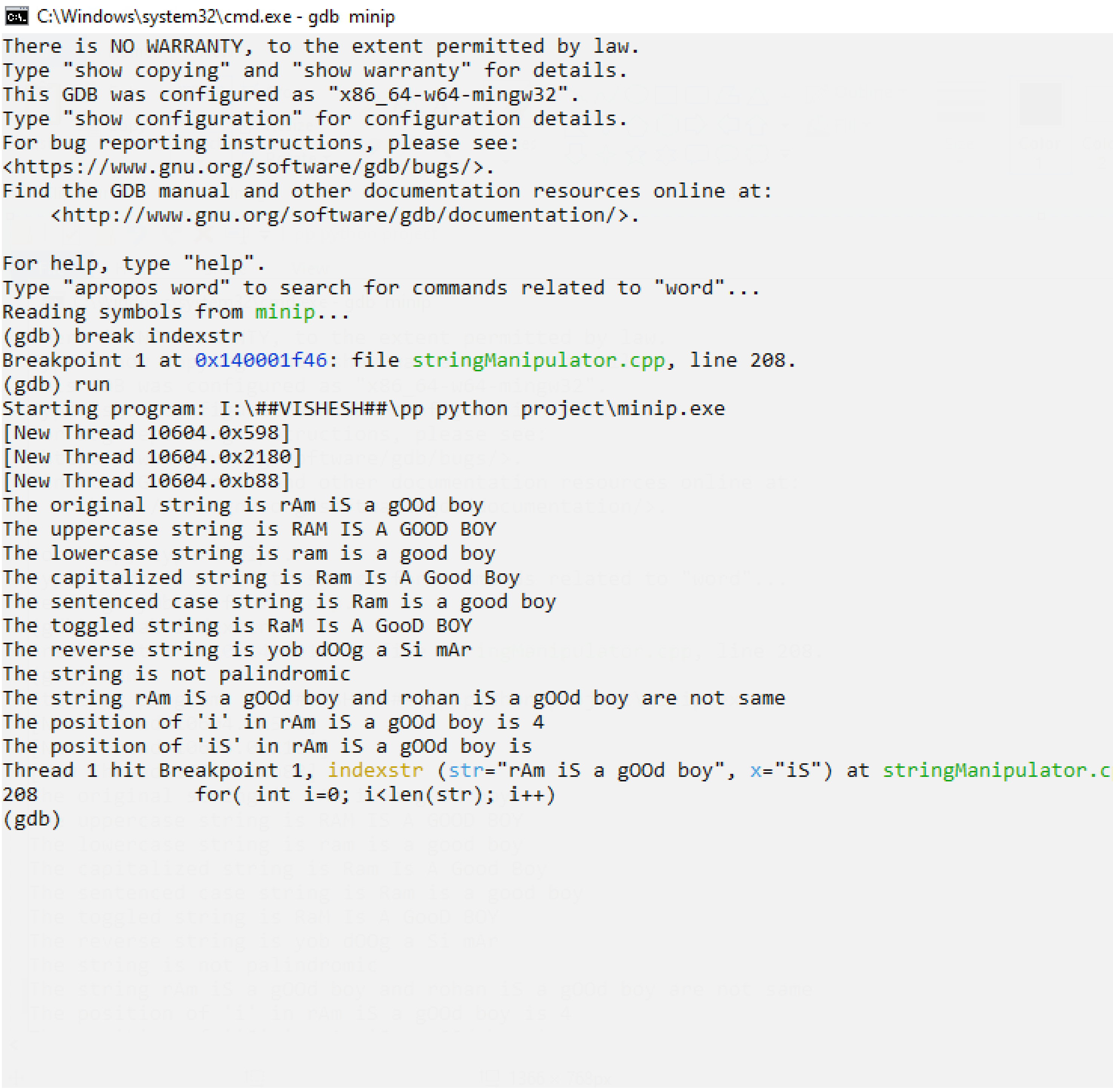
```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total           
 time   seconds   seconds    calls  Ts/call  Ts/call  name    

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
	   else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
	   function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
	   the function in the gprof listing. If the index is
	   in parenthesis it shows where it would appear in
	   the gprof listing if it were to be printed.
```

DEBBUGING STEPS

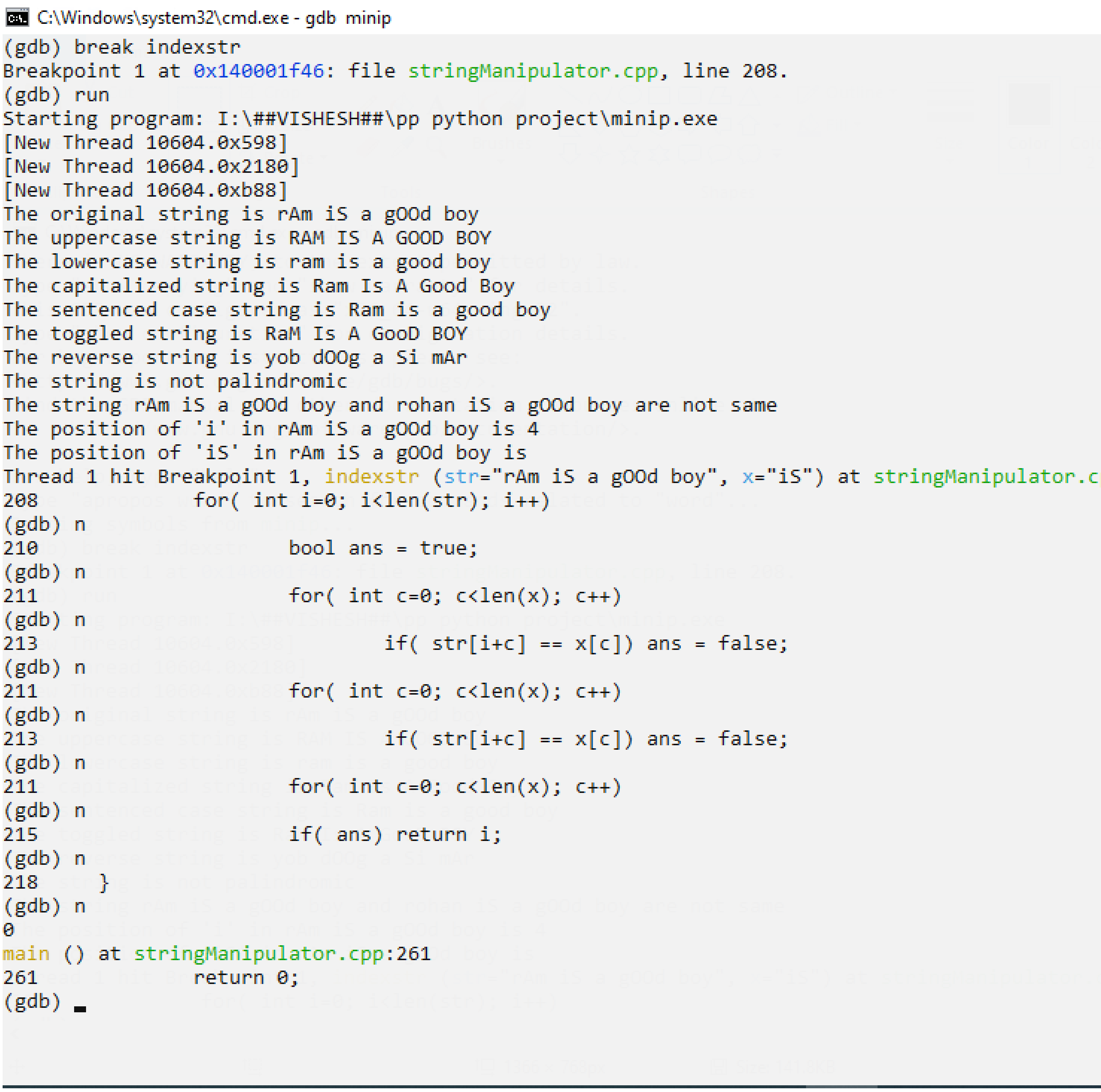


C:\Windows\system32\cmd.exe - gdb minip

```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Paridhi Educational>I:

I:\>cd ##VISHESH##\pp python project

I:\##VISHESH##\pp python project>g++ -g stringManipulator.cpp -o minip

I:\##VISHESH##\pp python project>gdb minip
GNU gdb (GDB) 11.2
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from minip...
(gdb)
```


C:\Windows\system32\cmd.exe - gdb minip

```
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-w64-mingw32".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from minip...
(gdb) break indexstr
Breakpoint 1 at 0x140001f46: file stringManipulator.cpp, line 208.
(gdb) run
Starting program: I:\##VISHESH##\pp python project\minip.exe
[New Thread 10604.0x598]
[New Thread 10604.0x2180]
[New Thread 10604.0xb88]
The original string is rAm iS a gOOd boy
The uppercase string is RAM IS A GOOD BOY
The lowercase string is ram is a good boy
The capitalized string is Ram Is A Good Boy
The sentenced case string is Ram is a good boy
The toggled string is RaM Is A GooD BOY
The reverse string is yob dOOg a Si mAr
The string is not palindromic
The string rAm iS a gOOd boy and rohan iS a gOOd boy are not same
The position of 'i' in rAm iS a gOOd boy is 4
The position of 'iS' in rAm iS a gOOd boy is
Thread 1 hit Breakpoint 1, indexstr (str="rAm iS a gOOd boy", x="iS") at stringManipulator.c
208             for( int i=0; i<len(str); i++)
(gdb)
```



```
(gdb) break indexstr
Breakpoint 1 at 0x140001f46: file stringManipulator.cpp, line 208.
(gdb) run
Starting program: I:\##VISHESH##\pp python project\minip.exe
[New Thread 10604.0x598]
[New Thread 10604.0x2180]
[New Thread 10604.0xb88]
The original string is rAm iS a gOOd boy
The uppercase string is RAM IS A GOOD BOY
The lowercase string is ram is a good boy
The capitalized string is Ram Is A Good Boy
The sentenced case string is Ram is a good boy
The toggled string is RaM Is A GooD BOY
The reverse string is yob dOOg a Si mAr
The string is not palindromic
The string rAm iS a gOOd boy and rohan iS a gOOd boy are not same
The position of 'i' in rAm iS a gOOd boy is 4
The position of 'iS' in rAm iS a gOOd boy is
Thread 1 hit Breakpoint 1, indexstr (str="rAm iS a gOOd boy", x="iS") at stringManipulator.c
208             for( int i=0; i<len(str); i++)
(gdb) n
210                     bool ans = true;
(gdb) n
211                     for( int c=0; c<len(x); c++)
(gdb) n
213                             if( str[i+c] == x[c]) ans = false;
(gdb) n
211                     for( int c=0; c<len(x); c++)
(gdb) n
213                             if( str[i+c] == x[c]) ans = false;
(gdb) n
211                     for( int c=0; c<len(x); c++)
(gdb) n
215                     if( ans) return i;
(gdb) n
218     }
(gdb) n
0
main () at stringManipulator.cpp:261
261             return 0;
(gdb)
```

MISCELLANEOUS DATA

Starting Date : 15 November, 2022
End Date : 15 November, 2022
Total time required : 2 hours
Total line of code : 262
No of functions : 17
Language used : C++
Profiller used : Gpof
Debugger used : gdb
Program Title : String Manipulator