**PROBLEM SOLVING**

(Solving Various Problems Using C & C++ Language)

*Summer Internship Report Submitted in partial fulfilment*

*of the requirement for under graduate degree of*

**Bachelor of Technology**

In

**Computer Science Engineering**

By

**VemulapalliRishitha**

**221710307062**

<https://github.com/rishitha7062>

*Under the Guidance of*

Assistant Professor



Department Of Computer Science Engineering

GITAM School of Technology

GITAM (Deemed to be University)

Hyderabad-502329

July 2020

**DECLARATION**

I submit this industrial training work entitled as **“Solving Various Problems Using C and C++ Language”** to GITAM (Deemed to Be University), Hyderabad in partial fulfilment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science Engineering**”. I declare that it was carried out independently by me under the guidance of  , GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD          VEMULAPALLI RISHITHA

Date: 20-07-2020                                                                                                      221710307062



GITAM (DEEMED TO BE UNIVERSITY)

     Hyderabad-502329, India

                                                                 Dated: 20-7-2020

**CERTIFICATE**

This is to certify that the Industrial Training Report entitled as **“Solving Various Problems Using C and C++ Language ”** is being submitted by VEMULAPALLIRISHITHA (221710307062) in partial fulfilment of the requirement for the award of **Bachelor of Technology in Computer Science Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic year 2019-20.

It is faithful record work carried out by her at the **Computer Science Engineering Department**, GITAM University Hyderabad Campus under my guidance and supervision.

**Dr.S. Phani Kumar**

Assistant Professor                                                              Assistant Professor and HOD

Department of CSE                                                         Department of CSE

**ACKNOWLEDGEMENT**

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad,** Pro Vice Chancellor, GITAM Hyderabad and **Dr.N. Seetharamaiah,** Principal, GITAM Hyderabad

I would like to thank respected **Dr.S. Phani Kumar,** Head of the Department of Computer Science Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a internship report. It helped me a lot to realize of what we study for.

I would like to thank the respected faculties who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

                                                                                    VEMULAPALLLI RISHITHA

                                                                                                 221710307062

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| 1. **INTRODUCTION** | **1** |
| 1. **PROBLEM 1-Reverse Gear**   2.1 problem statement  2.2 Coding  2.3 Output   1. **PROBLEM 2**- **Counting Rock Samples**   3.1 problem statement  3.2 Coding  3.3 Output   1. **PROBLEM 3**- **Philaland Coin Problem**   4.1 problem statement  4.2 Coding  4.3 Output   1. **PROBLEM 4**- **Super ASCII String Checker**   5.1 problem statement  5.2 Coding  5.3 Output   1. **PROBLEM 5**- **Cyclic palindrome**   6.1 problem statement  6.2 Coding  6.3 Output   1. **DESIGN REQUIREMENTS**   7.1 Hardware Requirements  7.2 Software Requirements  8.  **BIBLIOGRAPHY** | **3**  3  6  7  **3**  8  11  12  **13**  13  15  16  **17**  17  19  20  **21**  21  24  36  **27**  27  27  **28** |

# 1 Introduction

Problem solving is the act of defining a problem; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution.

There are five problems which are listed below are of different complexity and require different approach and logics in order to achieve desired Output/ Solution.

1. **Reverse Gear:** In this problem we calculate the time taken by the car to hit the wall based on given input and as per given condition.
2. **Counting Rock Samples:** In this problem we find out the number of rocks present in each of the ranges as per given condition and based on given input.
3. **Philaland Coin Problem:** In this problem we find out minimum number of denominations of coins required based on given input.
4. **Super ASCII String Checker**: In this problem we check whether the given string is a super ascii string or not as per given condition.
5. **Cyclic palindrome**: In this problem we find out the minimum number of cyclic shifts needed to make the given string, a palindrome based on given input and as per given condition.

C is a general-purpose high-level language that was originally developed by Dennis Ritchie for the Unix operating system. It was first implemented on the Digital Equipment Corporation PDP-11 computer in 1972. C was initially used for system development work, in particular the programs that make-up the operating system. C was adoped as a system development language because it produces code that runs nearly as fast as code written in assembly language.

C++ is an extension to C language and was developed by Bjarne Stroustrup at bell labs. C++ is an intermediate-level language, as it comprises a confirmation of both high level and low-level language features. C++ is a statically typed, free form, multiparadigm, compiled general-purpose language.

C++ is an Object Oriented Programming language but is not purely Object Oriented. Its features like 'Friend' and 'Virtual', violate some of the very important OOPS features, rendering this language unworthy of being called completely Object Oriented. It is a middle-level language.

I have executed projects in C and C++ language . For C and C++ language, I have used DEV C++ to execute the codes.

# 

# 2 Problem 1

**Reverse Gear**

This is a problem where we calculate the time taken by the car to hit the wall based on given input and as per given condition.

## 2.1 Problem Statement:-

A futuristic company is building an autonomous car. The scientists at the company are training the car to perform Reverse parking. To park, the car needs to be able to move in backward as well as forward direction. The car is programmed to move backwards B meters and forwards again, say F meters, in a straight line. The car does this repeatedly until it is able to park or collides with other objects. The car covers 1 meter in T units of time. There is a wall after distance D from car's initial position in the backward direction.

The car is currently not without defects and hence often hits the wall. The scientists are devising a strategy to prevent this from happening. Your task is to help the scientists by providing them with exact information on amount of time available before the car hits the wall.

**Input:-**

First line contains total number of test cases, denoted by N

Next N lines, contain a tuple containing 4 values delimited by space

F B T D, where

1. F denotes forward displacement in meters
2. B denotes backward displacement in meters
3. T denotes time taken to cover 1 meter
4. D denotes distance from Car's starting position and the wall in backward direction

**Output:-**

For each test case print time taken by the Car to hit the wall

**Constraints:-**

First move will always be in backward direction

1 <= N <= 100

backward displacement > forward displacement i.e. (B > F)

forward displacement (F) > 0

backward displacement (B) > 0

time (T) > 0

distance (D) > 0

All input values must be positive integers only

[**Sample Input and Output**](https://www.blogger.com/null)**:-**

|  |  |  |
| --- | --- | --- |
| [SNo.](https://www.blogger.com/null) | [Input](https://www.blogger.com/null) | [Output](https://www.blogger.com/null) |
| 1 | [2](https://www.blogger.com/null)  [6 9 3 18](https://www.blogger.com/null)  [3 7 5 20](https://www.blogger.com/null) | [162](https://www.blogger.com/null)  [220](https://www.blogger.com/null) |

**Concepts Used To Solve:-**

* **Arrays:** An array is a collection of data items, all of the same type, accessed using a common name. A one-dimensional array is like a list; A two dimensional array is like a table. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
* **While loop:** A while loop in C programming is a control flow statement that repeatedly executes a target statement as long as a given condition is true.

**SYNTAX:**

while(Condition)

**{**

Statement(s);

}

* **For loop:** A for loop in C programming is a control flow statement for specifying iteration, which allows code to be executed repeatedly a specific number of times.

**SYNTAX:**

for(init; Condition; increment/decrement)

**{**

Statement(s);

}

* **If-else statement:** If condition evaluates to true, the then-statement runs. If condition is false, the else-statement runs.

**SYNTAX:**

if (condition)

{

then-statement(s);

}

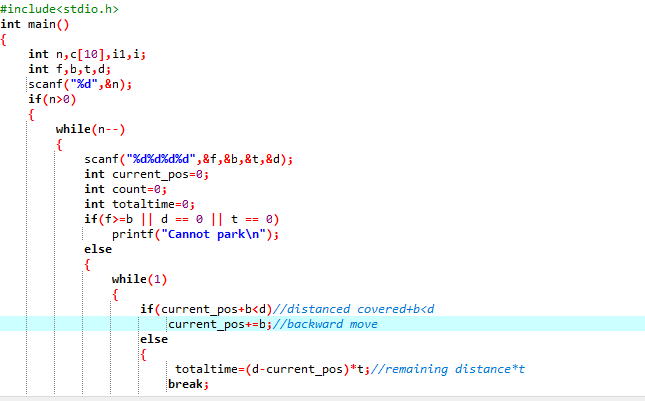
else

{

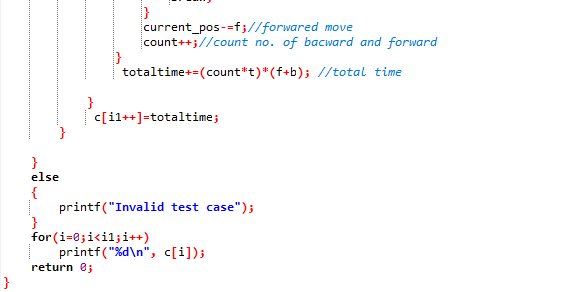
else-statement(s);

}

## 2.2 Coding

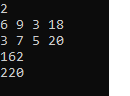


**Fig 2.2.1**

****

**Fig 2.2.2**

## 2.3 Output



**Fig 2.3.1**

# 3 Problem 2

# Counting Rock Samples

This is a problem where we find out the number of rocks present in each of the ranges as per given condition and based on given input.

## 3.1 Problem Statement:-

Juan Marquinho is a geologist and he needs to count rock samples in order to send it to a chemical laboratory. He has a problem: The laboratory only accepts rock samples by a range of its size in ppm (parts per million).Juan Marquinho receives the rock samples one by one and he classifies the rock samples according to the range of the laboratory. This process is very hard because the number of rock samples may be in millions.Juan Marquinho needs your help, your task is to develop a program to get the number of rocks in each of the ranges accepted by the laboratory

**Input:-**

An positive integer S (the number of rock samples) separated by a blank space, and a positive integer R (the number of ranges of the laboratory); A list of the sizes of S samples (in ppm), as positive integers separated by space R lines where the ith line containing two positive integers, space separated, indicating the minimum size and maximum size respectively of the ith range

**Output:-**

R lines where the ith line contains a single non-negative integer indicating the number of the samples which lie in the ith range.

**Constraints:-**

* 10 <= S <= 10000
* 1 <= R <= 1000000
* 1<=size of Sample <= 1000

[**Sample Input and Output**](https://www.blogger.com/null)**:-**

**Example 1:-**

**Input:** 10 2

* 345 604 321 433 704 470 808 718 517 811
* 300 350
* 400 700

**Output:** 2 4

**Explanation:** There are 10 samples (S) and 2 ranges (R). The samples are 345, 604,811. The ranges are 300-350 and 400-700. There are 2 samples in the first range (345 and 321) and 4 samples in the second range (604, 433, 470, 517). Hence the two lines of the output are 2 and 4

**Example 1:-**

**Input:** 20 3

* 921 107 270 631 926 543 589 520 595 93 873 424 759 537 458 614 725 842 575 195
* 1 100
* 50 600
* 1 1000

**Output:** 1 12 20

**Explanation:** There are 20 samples and 3 ranges. The samples are 921, 107 195. The ranges are 1-100, 50-600 and 1-1000. Note that the ranges are overlapping. The number of samples in each of the three ranges

**Concepts Used To Solve:-**

* **Arrays:** An array is a collection of data items, all of the same type, accessed using a common name. A one-dimensional array is like a list; A two dimensional array is like a table. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
* **For loop:** A for loop in C programming is a control flow statement for specifying iteration, which allows code to be executed repeatedly a specific number of times.

**SYNTAX:**

for(init; Condition; increment/decrement)

**{**

Statement(s);

}

* **If statement:** If condition evaluates to true, the then-statement runs.

**SYNTAX:**

if (condition)

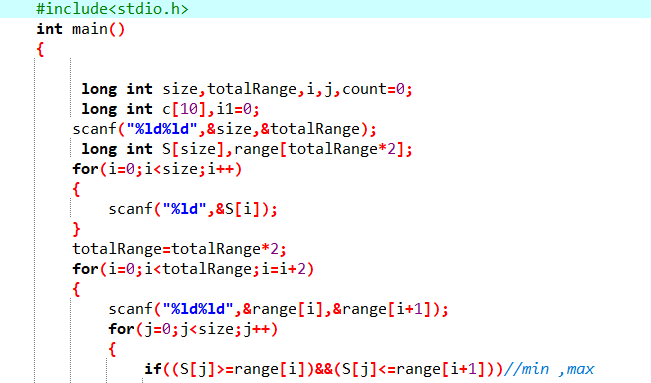
{

then-statement(s);

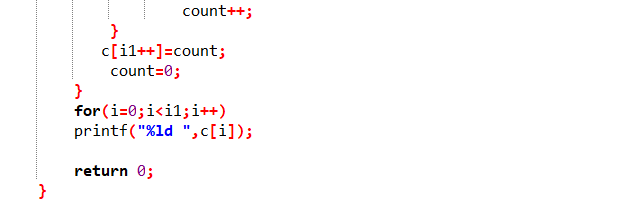
}

* **Datatype – longint(**contain integer numbers that are too large to fit in the Integer datatype and holds signed 64-bit)

## 3.2 Coding

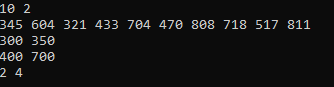


**Fig 3.2.1**

****

**Fig 3.2.2**

## 3.3 Output



**Fig 3.3.1**

# 

# 4 Problem 3

# Philaland Coin Problem

This is a problem where we find out minimum number of denominations of coins required based on given input.

## 4.1 Problem Statement:-

The problem solvers have found a new Island for coding and named it as Philaland.These smart people were given a task to make purchase of items at the Island easier by distributing various coins with different value.Manish has come up with a solution that if we make coins category starting from $1 till the maximum price of item present on Island, then we can purchase any item easily. He added following example to prove his point.

Let’s suppose the maximum price of an item is 5$ then we can make coins of {$1, $2, $3, $4, $5}to purchase any item ranging from $1 till $5.

Now Manisha, being a keen observer suggested that we could actually minimize the number of coins required and gave following distribution {$1, $2, $3}. According to her any item can be purchased one time ranging from $1 to $5. Everyone was impressed with both of them.Your task is to help Manisha come up with minimum number of denominations for any arbitrary max price in Philaland.

**Input:-**

First line contains an integer T denoting the number of test cases.

Next T lines contains an integer N denoting the maximum price of the item presents Philaland.

**Output:-**

For each test case print a single line denoting the minimum number of denominations of coins required.

**Constraints:-**

* 1<=T<=100
* 1<=N<=5000

**Concepts Used To Solve:-**

* **Arrays:** An array is a collection of data items, all of the same type, accessed using a common name. A one-dimensional array is like a list; A two dimensional array is like a table. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
* **While loop:** A while loop in C programming is a control flow statement that repeatedly executes a target statement as long as a given condition is true.

**SYNTAX:**

while(Condition)

**{**

Statement(s);

}

[**Sample Input and Output**](https://www.blogger.com/null)**:**

|  |  |  |
| --- | --- | --- |
| [SNo.](https://www.blogger.com/null) | [Input](https://www.blogger.com/null) | [Output](https://www.blogger.com/null) |
| 1 | [2](https://www.blogger.com/null)  10  5 | 4  3 |

**Explanation:-**

**For test case 1**, N=10.

According to Manish {$1, $2, $3,… $10} must be distributed.

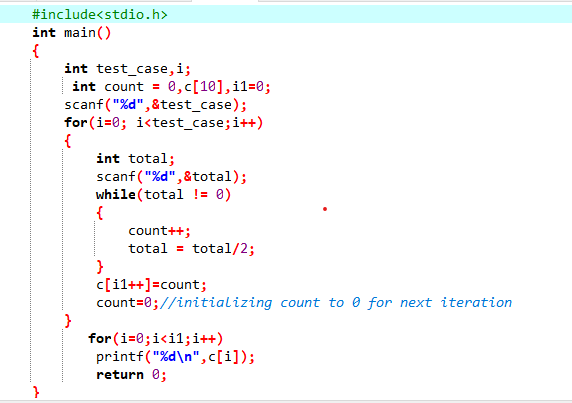
But as per Manisha only {$1, $2, $3, $4} coins are enough to purchase any item ranging from $1 to $10. Hence minimum is 4. Likewise denominations could also be {$1, $2, $3, $5}. Hence answer is still 4.

**For test case 2**, N=5.

According to Manish {$1, $2, $3, $4, $5} must be distributed.

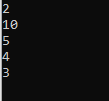
But as per Manisha only {$1, $2, $3} coins are enough to purchase any item ranging from $1 to $5. Hence minimum is 3. Likewise denominations could also be {$1, $2, $4}. Hence answer is still 3.

## 4.2 Coding



**Fig 4.2.1**

**4.3 Output**



**Fig 4.3.1**

# 5 Problem 4

**Super ASCII String Checker**

This is a problem where we check whether the given string is a super ascii string or not as per given condition.

## 5.1 Problem Statement:-

In the Byteland country a string "S" is said to super ascii string if and only if count of each character in the string is equal to its ascii value.

In the Byteland country ascii code of 'a' is 1, 'b' is 2 ...'z' is 26.

Your task is to find out whether the given string is a super ascii string or not.

**Input:-**

First line contains number of test cases T, followed by T lines, each containing a string "S”.

**Output:-**

For each test case print "Yes" if the String "S" is super ascii, else print "No"

**Constraints:-**

* 1<=T<=100
* 1<=|S|<=400, S will contains only lower case alphabets

**Concepts Used To Solve:-**

* **Arrays:** An array is a collection of data items, all of the same type, accessed using a common name. A one-dimensional array is like a list; A two dimensional array is like a table. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
* **While loop:** A while loop in C programming is a control flow statement that repeatedly executes a target statement as long as a given condition is true.

**SYNTAX:**

while(Condition)

**{**

Statement(s);

}

* **For loop:** **:** A for loop in C programming is a control flow statement for specifying iteration, which allows code to be excuted repeatedly a specific number of times.

**SYNTAX:**

for(init;Condition;increment/decrement)

**{**

Statement(s);

}

* **if-else statement:** If condition evaluates to true, the then-statement runs. If condition is false, the else-statement runs.

**SYNTAX:**

if (condition)

{

then-statement(s);

}

else

{

else-statement(s);

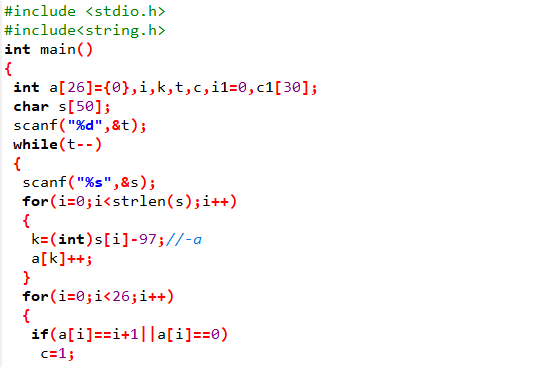
}

* **ASCII:** ASCII is the acronym for the *American Standard Code for Information Interchange.* It is a code for representing 128 English characters as numbers, with each letter assigned a number from 0 to 127.

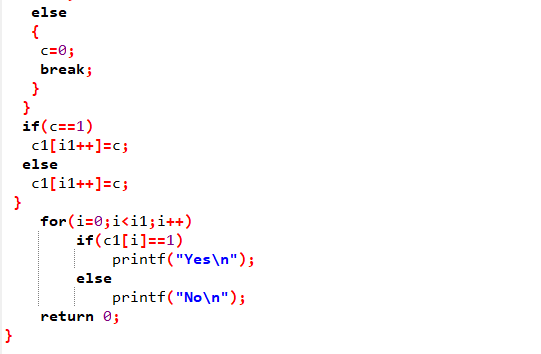
[**Sample Input and Output**](https://www.blogger.com/null)**:-**

|  |  |  |
| --- | --- | --- |
| [SNo.](https://www.blogger.com/null) | [Input](https://www.blogger.com/null) | [Output](https://www.blogger.com/null) |
| 1 | [2](https://www.blogger.com/null)  bba  scca | Yes  No |

## 5.2 Coding



**Fig 5.2.1**

****

**Fig 5.2.2**

## 5.3 Output



**Fig 5.3.1**

# 

# 6 Problem 5

**Cyclic palindrome**

This is a problem where we find out the minimum number of cyclic shifts needed to make the given string, a palindrome based on given input and as per given condition.

## 6.1 Problem Statement:-

A string is said to be palindrome, if it reads the same from both the ends. Given a string S, you are allowed to perform cyclic shifts. More formally, you can pick any one character from any end (head or tail) and you can append that character at the other end. For example, if the string is "abc", then if we do a shift using the character at head position then the string becomes "bca". Similarly, if we do the shift using the character at the tail then the input string becomes "cab". Your task is to find out the minimum number of shifts needed to make the given string, a palindrome. In case, we can't convert the string to palindrome then print -1.

**Input:-**

First line starts with T i.e. number of test cases, and then T lines will follow each containing a string "S".

**Output:-**

Print the minimum number of cyclic shifts for each string if it can be made a palindrome, else -1.

**Constraints:-**

* 1<=T<=100
* 1<=|S|<=300, S will contains only lower case alphabets ('a'-'z')

**Concepts Used To Solve:-**

* **Arrays:** An array is a collection of data items, all of the same type, accessed using a common name. A one-dimensional array is like a list; A two dimensional array is like a table. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
* **String function: str.size()**

**str.size():**this function is used to return the length of the string in terms of bytes

* **For loop:** **:** A for loop in C programming is a control flow statement for specifying iteration, which allows code to be excuted repeatedly a specific number of times.

**SYNTAX:**

for(init;Condition;increment/decrement)

**{**

Statement(s);

}

* **if-else statement:** If condition evaluates to true, the then-statement runs. If condition is false,the else-statement runs.

**SYNTAX:**

if (condition)

{

then-statement(s);

}

else

{

else-statement(s);

}

* **Break statement:** break statement is a loop control statement which is used to terminate the loop

**SYNTAX:**

break;

[**Sample Input and Output**](https://www.blogger.com/null)**:-**

|  |  |  |
| --- | --- | --- |
| [SNo.](https://www.blogger.com/null) | [Input](https://www.blogger.com/null) | [Output](https://www.blogger.com/null) |
| 1 | 4  abbb  aaabb  aabb  abc | -1  1  1  -1 |

**Explanation:-**

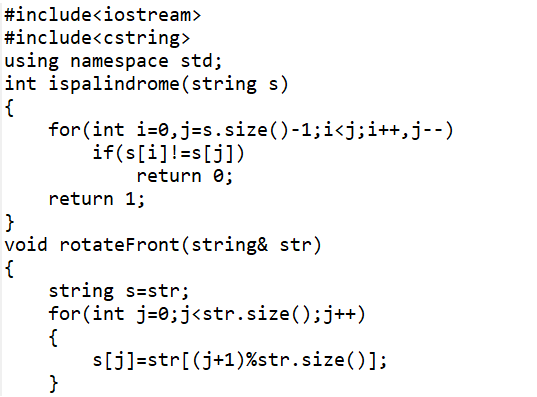
**For Test Case 2** (aaabb):

Shift the character at the tail to the head and the result will be "baaab", which is a palindrome. This is an operation which requires minimum number of shifts to make the given string a palindrome.

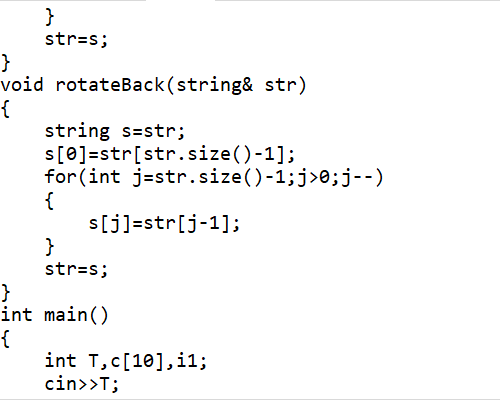
**For Test Case 3** (aabb):

One way to convert the given string to palindrome is, shift the character at the head to the tail, and the result will be "abba", which is a palindrome. Another way is to shift the character at the tail to the head, and the result will be "baab", which is also a palindrome. Both require only one shift.

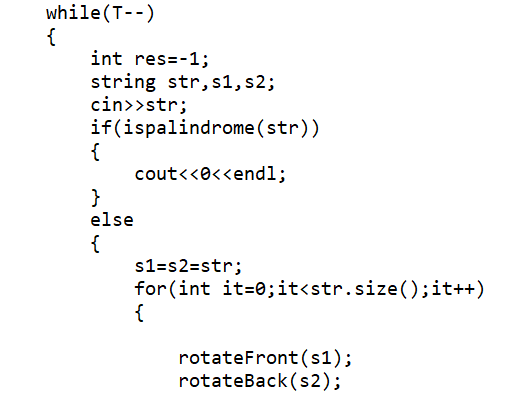
## 6.2 Coding



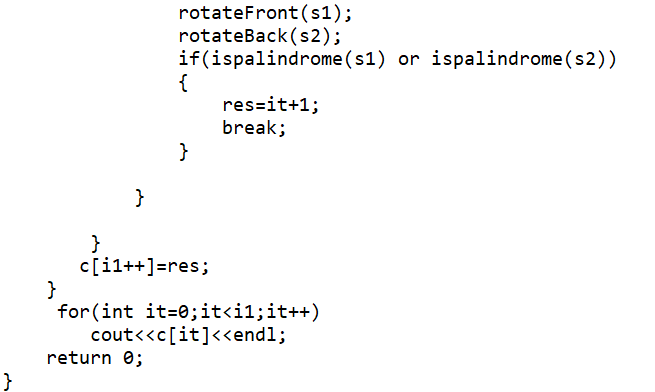
**Fig 6.2.1**

****

**Fig 6.2.2**

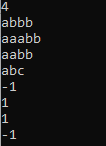
****

**Fig 6.2.3**

****

**Fig 6.2.4**

## 6.3 Output



**Fig 6.3.1**

# 7 Design Requirements

## 7.1 Hardware Requirements:-

This project can be executed in any system or an android phone without prior to any platform.

We can use any online compiler and interpreter.

## 7.2 Software Requirements:-

There are two ways to execute this project

1. Online compilers
2. Softwares for execution (DEV C++,…..)

Online Compilers require only internet connection. We have many free compilers with which we can code.

Softwares for execution need to be installed based on the user’s system specification. These help us to completely execute the project. These software are based on the platforms

**BIBLIOGRAPHY**

1. Vikash.“CodeVita Season IV Round 1 Question**:** Reverse Gear” *programminggeek.in*. Programming Geek, 8.Aug. 2015.
2. PrepInsta.“CodeVita: CountingRockSamples”prepinsta.com.PrepInsta,2.July. 2020.
3. Code of Geeks.“PhilalandCoin” *codeofgeeks.com*. CODE OF GEEKS, 19.Mar. 2020.
4. RajeevReddyNareddula”[TCS]SuperASCIIStringChecker(CodeVita)Problem”*matterhere.com.*Matterhere,2.July.2018.
5. Vikash.“TCSCodeVita2014Questions:Round2(Set2):CyclicPalindrome”programminggeek.in. Programming Geek, 26.Nov. 2014.