

PROBLEM SOLVING

(Solving various problems using C Language)

Summer Internship Report Submitted in partial fulfillment

of the requirement for under graduate degree of

Bachelor of Technology

In

Computer science and Engineering

By

B.Naresh

221710313004

<https://github.com/Naresh004/Adv-prg>

Under the Guidance of

Assistant Professor



Department Of Computer science and Engineering

GITAM School of Technology
GITAM (Deemed to be University)
Hyderabad-502329

July 2020

DECLARATION

I submit this industrial training work entitled “ **SOLVING VARIOUS PROBLEMS USING C LANGUAGE**” to GITAM (Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer science and Engineering**”. I declare that it was carried out independently by me under the guidance of -----, Asst. Professor, 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

B.NARESH

Date:

2217103130004



GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India

Dated:

CERTIFICATE

This is to certify that the Industrial Training Report entitled “**PROBLEM SOLVING**” is being submitted by B.NARESH(221710313004) in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer science and Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic year 2020-21

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

Mr. .Venkateswarlu

Assistant Professor

Department of CSE

Dr.K.Manjunathachari

Professor and HOD

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 completion of this internship.

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

I would like to thank respected **Dr. K. Manjunathachari**, Head of the Department of Electronics and Communication 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 **Mr. M. Venkateswarlu** 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.

B.NARESH
221710313004

TABLE OF CONTENTS

1 Introduction to the project	2
2 Problem 1 - Library management system	3
2.1 Problem Statement	3
2.2 Coding	6
2.3 Output	8
3 Problem 2 - Counting rock samples	11
3.1 Problem Statement	11
3.2 Coding	14
3.3 Output	15
4 Problem 3 - Milk man problem	16
4.1 Problem Statement	16
4.2 Coding	18
4.3 Output	19
5 Problem 4 - Consecutive prime sum	20
5.1 Problem Statement	20
5.2 Coding	22
5.3 Output	23
6 Problem 5 - Collecting candies	24
6.1 Problem Statement	24
6.2 Coding	27
6.3 Output	28
7 Software Requirements	29
7.1 Hardware Requirements	29
7.2 Software Requirements	29
8 References	30

1 Introduction to the project

Problem Solving is the Process of Designing and carrying out certain steps to reach a Solution. There are five problems which are listed below are of different complexity and require different approach and logic in order to achieve desired Output/ Solution.

1. **Library management system** - In this problem we add book information about the books such as author name, book name and finally we will display the information about books.
2. **Counting Rock Samples** - In this problem we find the number of rocks in each of the ranges accepted by the laboratory.
3. **Milk Man and His Bottles** - In this problem we find out the minimum number of bottles required to supply the given demand of milk.
4. **Consecutive Prime Sum** - In this problem we find out the number of prime numbers that satisfy the mentioned property in a given range.
($5 = 2 + 3$, $17 = 2 + 3 + 5 + 7$, $41 = 2 + 3 + 5 + 7 + 11 + 13$)
5. **Collecting Candies** - In this problem we calculate the minimum time in which all the candies can be collected.

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

2. PROBLEM-1 LIBRARY MANAGEMENT SYSTEM

2.1 Problem Statement:-

C PROGRAM ON LIBRARY MANAGEMENT SYSTEM.

Program to perform following actions :

1. Add book information.
2. Display book information.
3. List all books of given author.
4. List the title of specified book.
5. List the count of books in the library.

Concepts used for solving the problem :

Structures:

A structure is a user defined data type in C/C++. A structure creates a data type that can be used to group items of possibly different types into a single type.

Creation of structures:

‘struct’ keyword is used to create a structure. Following is an example.

```
struct address
{
    char name[50];
    char street[100];
    char city[50];
    char state[20];
    int pin;
};
```

Declaration of structures:

A structure variable can either be declared with structure declaration or as a separate declaration like basic types.


```
// A variable declaration with structure declaration.
struct Point
{
    int x, y;
} p1; // The variable p1 is declared with 'Point'

// A variable declaration like basic data types
struct Point
{
    int x, y;
};

int main()
{
    struct Point p1; // The variable p1 is declared like a normal variable
}
```

Initialization of structure members:

Structure members **cannot be** initialized with declaration. For example the following C program fails in compilation.

```
    struct Point
    {
        int x = 0; // COMPILER ERROR: cannot initialize members here
        int y = 0; // COMPILER ERROR: cannot initialize members here
    };
```

The reason for above error is simple, when a datatype is declared, no memory is allocated for it. Memory is allocated only when variables are created.

Structure members **can be** initialized using curly braces '{}'. For example, following is a valid initialization.

```
struct Point
{
    int x, y;
};

int main()
{
    // A valid initialization. member x gets value 0 and y
    // gets value 1. The order of declaration is followed.
    struct Point p1 = {0, 1};
}
```

Accessing structure elements:

Structure members are accessed using dot (.) operator.

```
#include<stdio.h>

struct Point
{
    int x, y;
};

int main()
{
    struct Point p1 = {0, 1};

    // Accessing members of point p1
    p1.x = 20;
    printf("x = %d, y = %d", p1.x, p1.y);

    return 0;
}
```

Output:

```
x = 20, y = 1
```

Arrays:

An array in C or C++ is a collection of items stored at contiguous memory locations and elements can be accessed randomly using indices of an array. They are used to store similar type of elements as in the data type must be the same for all elements. They can be used to store collection of primitive data types such as int, float, double, char, etc of any particular type. To add to it, an array in C or C++ can store derived data types such as the structures, pointers etc. Given below is the picturesque representation of an array.

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

Array Length = 9
First Index = 0
Last Index = 8

Fig 2.1.1

2.2 Coding:

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct library
{
char bk_name[30];
char author[30];
int pages;
float price;
};

int main()
{
struct library l[100];
char ar_nm[30],bk_nm[30];
int i,j, keepcount;
i=j=keepcount = 0;

while(j!=6)
{
printf("\n\n1. Add book information\n2. Display book information\n");
printf("3. List all books of given author\n");
printf("4. List the title of specified book\n");
printf("5. List the count of books in the library\n");
printf("6. Exit");

```

```

printf ("\n\nEnter one of the above : ");
scanf ("%d",&j);

switch (j)
{
    /* Add book */
    case 1:

        printf ("Enter book name = ");
        scanf ("%s",l[i].bk_name);

        printf ("Enter author name = ");
        scanf ("%s",l[i].author);

        printf ("Enter pages = ");
        scanf ("%d",&l[i].pages);

        printf ("Enter price = ");
        scanf ("%f",&l[i].price);
        keepcount++;
        i++;
        break;

    case 2:
        printf ("\nyou have entered the following information\n");
        for(i=0; i<keepcount; i++)
        {
            printf ("\nbook name = %s",l[i].bk_name);

            printf ("\nauthor name = %s",l[i].author);

            printf ("\npages = %d",l[i].pages);

            printf ("\nprice = %f",l[i].price);
        }
        break;

    case 3:
        printf ("Enter author name : ");
        scanf ("%s",ar_nm);
        for (i=0; i<keepcount; i++)
        {
            if (strcmp(ar_nm, l[i].author) == 0)
                printf ("%s %s %d %f",l[i].bk_name,l[i].author,l[i].pages,l[i].price);
        }
        break;
}

```

```

case 4:
printf ("Enter book name : ");
scanf ("%s",bk_nm);
for (i=0; i<keepcount; i++)
{
if (strcmp(bk_nm, l[i].bk_name) == 0)
printf ("%s \t %s \t %d \t %f",l[i].bk_name,l[i].author,l[i].pages,l[i].price);
}
break;

case 5:
printf("\n No of books in library : %d", keepcount);
break;
case 6:
exit (0);

}
}
return 0;
}

```

2.3 OUTPUT:

```

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

Enter one of the above : 1
Enter book name = c-programming
Enter author name = balaguruswamy
Enter pages = 500
Enter price = 400

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

Enter one of the above : 1
Enter book name = maths
Enter author name = rdsharma
Enter pages = 900
Enter price = 600

```

Fig-2.3.1

```
1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

Enter one of the above : 2

you have entered the following information

book name = c-programming
author name = balaguruswamy
pages = 500
price = 400.000000
book name = maths
author name = rdsharma
pages = 900
price = 600.000000

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

Enter one of the above : 3
Enter author name : balaguruswamy
c-programming balaguruswamy 500 400.000000
```

Fig-2.3.2

```

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

Enter one of the above : 4
Enter book name : maths
maths    rdsharma      900      600.000000

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

Enter one of the above : 5

No of books in library : 2

1. Add book information
2. Display book information
3. List all books of given author
4. List the title of specified book
5. List the count of books in the library
6. Exit

```

Fig-2.3.3

```

Enter one of the above : 6

-----
Process exited after 43.47 seconds with return value 0
Press any key to continue . . .

```

Fig-2.3.4

3. PROBLEM-2

Counting Rock Samples

3.1 Problem Statement:-

Explanation :

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 Format:

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 Format:

R lines where the ith line containing 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 each sample (in ppm) ? 1000

Expected Output:

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.

Concepts used to solve :-

For loop and if else concepts are used.

For loop :

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax :

The syntax of a for loop in C programming language is –

```
for (initializationStatement; testExpression; updateStatement)
{
    // statements inside the body of loop
}
```

If-else Statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.

```
if(expression){  
  //code to be executed if condition is true  
}else{  
  //code to be executed if condition is false  
}
```

3.2 Coding:

```
#include<stdio.h>
int main() {
int a[1000],s,i,j,t,l1,l2,c=0;
scanf("%d",&s);
scanf("%d",&t);
for(i=0;i<s;i++)
scanf("%d",&a[i]);
for(i=0;i<t;i++)
{
scanf("%d %d",&l1,&l2);
for(j=0;j<s;j++)
{
if((a[j]>=l1)&&(a[j]<=l2))
c++;
}
printf("%d\n",c);
c=0;
}
return 0;
}
```

3.3 OUTPUT:

```
10 2
345 604 321 433 704 470 808 718 517 811
300 350
2
400 700
4

-----
Process exited after 29.84 seconds with return value 0
Press any key to continue . . .
```

Fig-3.3.1

4. PROBLEM-3

Milk Man and His Bottles

4.1 Problem Statement:

Explanation :

A Milkman serves milk in packaged bottles of varied sizes. The possible size of the bottles are {1, 5, 7 and 10} liters. He wants to supply desired quantity using as less bottles as possible irrespective of the size. Your objective is to help him find the minimum number of bottles required to supply the given demand of milk.

Input Format:

First line contains number of test cases N

Next N lines, each contain a positive integer L_i which corresponds to the demand of milk.

Output Format:

For each input L_i , print the minimum number of bottles required to fulfill the demand.

Constraints:

$1 \leq N \leq 1000$

$L_i > 0$

$1 \leq i \leq N$

Sample Input and Output :

S.No.	Input	Output
1	2	
	17	2
	65	7

Explanation:

Number of test cases is 2

1. In first test case, demand of milk is 17 liters which can be supplied using minimum of 2 bottles as follows :

- 1 x 10 liters and
- 1 x 7 liters

2. In second test case, demand of milk is 65 liters which can be supplied using minimum of 7 bottles as follows :

- 6 x 10 liters and
- 1 x 5 liters.

Concepts used to solve :-

For loop and if else concepts are used.

For loop :

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax :

The syntax of a for loop in C programming language is –

```
for (initializationStatement; testExpression; updateStatement)
{
    // statements inside the body of loop
}
```

If-else Statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.

```
if(expression){
    //code to be executed if condition is true
}else{
    //code to be executed if condition is false
}
```

4.2 CODING :

```
#include<stdio.h>
int main()
{
    int n,a[20],c[20]={0},i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        while(a[i]!=0)
        {
            if(a[i]>=10)
            {
                c[i]+=a[i]/10;
                a[i]=a[i]%10;;
            }
            else if(a[i]>=7)
            {
                c[i]+=a[i]/7;
                a[i]=a[i]%7;;
            }
            else if(a[i]>=5)
            {
                c[i]+=a[i]/5;
                a[i]=a[i]%5;;
            }
            else if(a[i]<5)
            {
                c[i]+=a[i]/1;
                a[i]=a[i]%1;;
            }
        }
    }
    for(i=0;i<n;i++)
    {
        printf("%d\n",c[i]);
    }
    return 0;
}
```

4.3 OUTPUT :

```
2
17
65
2
7
-----
Process exited after 11.19 seconds with return value 0
Press any key to continue . . .
```

Fig-4.3.1

5. PROBLEM-4

Consecutive Prime Sum

5.1 Problem Statement:-

Some prime numbers can be expressed as a sum of other consecutive prime numbers. For example $5 = 2 + 3$, $17 = 2 + 3 + 5 + 7$, $41 = 2 + 3 + 5 + 7 + 11 + 13$. Your task is to find out how many prime numbers which satisfy this property are present in the range 3 to N subject to a constraint that summation should always start with number 2.

Write code to find out the number of prime numbers that satisfy the above-mentioned property in a given range.

S.no	Input	Output	Comment
1	20	2	(Below 20, there are 2 such members: 5 and 17) $5 = 2+3$ $17 = 2+3+5+7$
2	15	1	

Input Format: First line contains a number N

Output Format: Print the total number of all such prime numbers which are less than or equal to N.

Constraints: $2 < N \leq 12,000,000,000$

Concepts used to solve :-

For loop and if else concepts are used.

For loop :

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax :

The syntax of a for loop in C programming language is –

```
for (initializationStatement; testExpression; updateStatement)
{
    // statements inside the body of loop
}
```

If-else Statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.

```
if(expression){
//code to be executed if condition is true
}else{
//code to be executed if condition is false
}
```

5.2 Coding:

```
#include <stdio.h>
int prime(int b)
{
    int j,cnt;
    cnt=1;
    for(j=2;j<=b/2;j++)
    {
        if(b%j==0)
            cnt=0;
    }
    if(cnt==1)
        return 1;
    else
        return 0;
}

int main() {
    int i,j,n,cnt,a[25],c,sum=0,count=0,k=0;
    scanf("%d",&n);
    for(i=2;i<=n;i++)
    {
        cnt=1;
        for(j=2;j<=n/2;j++)
        {
            if(i%j==0)
                cnt=0;
        }
        if(cnt==1)
        {
            a[k]=i;
            k++;
        }
    }
}
```

```

for(i=0;i<k;i++)
{
    sum=sum+a[i];
    c= prime(sum);
    if(c==1)
        count++;
}
printf("%d",count);
return 0;
}

```

5.3 OUTPUT :

```

20
2
-----
Process exited after 12.18 seconds with return value 0
Press any key to continue . . .

```

Fig-5.3.1

6. PROBLEM-5

Collecting Candies

6.1 Problem Statement:- :

Krishna loves candies a lot, so whenever he gets them, he stores them so that he can eat them later whenever he wants to.

He has recently received N boxes of candies each containing C_i candies where C_i represents the total number of candies in the i th box. Krishna wants to store them in a single box. The only constraint is that he can choose any two boxes and store their joint contents in an empty box only. Assume that there are infinite number of empty boxes available.

At a time he can pick up any two boxes for transferring and if both the boxes say contain X and Y number of candies respectively, then it takes him exactly $X+Y$ seconds of time. As he is too eager to collect all of them he has approached you to tell him the minimum time in which all the candies can be collected.

Input Format:

- The first line of input is the number of test case T
- Each test case is comprised of two inputs
- The first input of a test case is the number of boxes N
- The second input is N integers delimited by whitespace denoting the number of candies in each box

Output Format:

Print minimum time required, in seconds, for each of the test cases. Print each output on a new line.

Constraints:

- $1 \leq T \leq 10$
- $1 \leq N \leq 10000$
- $1 \leq [\text{Candies in each box}] \leq 100009$

Sample Input and Output:

Sample	Input	Output
1	1 4 1 2 3 4	19
2	1 5 1 2 3 4 5	34

Explanation for sample input-output 1:

4 boxes, each containing 1, 2, 3 and 4 candies respectively. Adding 1 + 2 in a new box takes 3 seconds. Adding 3 + 3 in a new box takes 6 seconds. Adding 4 + 6 in a new box takes 10 seconds. Hence total time taken is 19 seconds. There could be other combinations also, but overall time does not go below 19 seconds.

Explanation for sample input-output 2:

5 boxes, each containing 1, 2, 3, 4 and 5 candies respectively. Adding 1 + 2 in a new box takes 3 seconds. Adding 3 + 3 in a new box takes 6 seconds. Adding 4 + 6 in a new box takes 10 seconds. Adding 5 + 10 in a new box takes 15 seconds. Hence total time taken is 34 seconds. There could be other combinations also, but overall time does not go below 33 seconds.

Concepts used to solve :-

For loop and if else concepts are used.

For loop :

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax :

The syntax of a for loop in C programming language is –

```
for (initializationStatement; testExpression; updateStatement)
{
```

```
// statements inside the body of loop  
}
```

If-else Statement

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition. The syntax of the if-else statement is given below.

```
if(expression){  
  //code to be executed if condition is true  
}else{  
  //code to be executed if condition is false  
}
```

6.2 CODING:

```
#include <stdio.h>
int main() {
    int n,i,k=0,sum=0,s1=0,t,temp=0,j,l;
    long c[100009],s[100009];
    scanf("%d",&t);
    for(l=0;l<t;l++)
    {
        scanf("%d",&n);
        for(i=0;i<n;i++)
            scanf("%ld",&c[i]);
        for(i=0;i<n;i++)
        {
            for(j=i+1;j<n;j++)
            {
                if(c[i]>c[j])
                {
                    temp=c[i];
                    c[i]=c[j];
                    c[j]=temp;
                }
            }
        }
        sum=0;
        k=0;
        for(i=0;i<n;i++)
        {
            sum=sum+c[i];
            s[k]=sum;
            k++;
        }
        s1=0;
        for(i=1;i<k;i++)
            s1=s1+s[i];
        printf("%d",s1);
    }
    return 0;
}
```


6.3 OUTPUT :

```
1
4
1 2 3 4
19
-----
Process exited after 9.835 seconds with return value 0
Press any key to continue . . .
```

Fig-6.3.1

```
1
5
1 2 3 4 5
34
-----
Process exited after 12.3 seconds with return value 0
Press any key to continue . . .
```

Fig-6.3.2

7. Software 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 projects

1. Online compilers.
2. Software for execution (DEV C++).

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

Software 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

8. REFERENCES

- <https://www.faceprep.in/tcs/tcs-codevita-questions/>
- <https://prepinsta.com/tcs-codevita/practice-questions-with-answers/>
- <https://www.geeksforgeeks.org/>