

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

DHARMAVARAPU PRAVALLIKA

221710307021

<https://github.com/pravallika00/Adv-Prog>

Under the Guidance of



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 & 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

D.Pravallika

Date: 20-07-2020

221710307021



GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India

Dated: 20-07-2020

CERTIFICATE

This is to certify that the Industrial Training Report entitled “**Solving various problems with C language**” is being submitted by D.PRAVALLIKA (221710307021) in partial fulfillment 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.

Assistant Professor
Department of CSE

Mr.S.Phani Kumar
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 competition of this internship.

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

I would like to thank respected Mr.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.

D.Pravallika
221710307021

TABLE OF CONTENTS

1 Introduction to the project	7
2 Problem 1 - Philaland coin problem	8
2.1 Problem Statement	8
2.2 Coding	10
2.3 Output	10
3 Problem 2 - The Vita Sum	11
3.1 Problem Statement	11
3.2 Coding	12
3.3 Output	12
4 Problem 3 - Reverse Gear	13
4.1 Problem Statement	14
4.2 Coding	15
4.3 Output	15
5 Problem 4 - Consecutive Prime Sum	16
5.1 Problem Statement	16
5.2 Coding	17
5.3 Output	18
6 Hardware Requirements	19
7 Software Requirements	19

1 Introduction to the project

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

1. **Philaland coin problem** - This is a problem where we are going to make purchase of items at the Island easier by distributing various coins with different value.
2. **The Vita Sum** - In this problem where we are going to find out the sum of all even combination of balls that can taken out of the bag.
3. **Reverse Gear** - In this problem the car needs to be able to move in backward as well as forward direction.
4. **Consecutive Prime Sum** - In this problem we have to find out the number of prime numbers that satisfy the given ranges.

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

2 Problem 1

Philaland Coin Problem

2.1 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 him 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 Format

- 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 present Philaland.

Output Format

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

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 5000$

Sample Input

2
10
5

Sample Output

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.

Concepts Used To Solve:-

While loop : It is used to repeat a section of code an unknown number of times until a specific condition is met.

2.2 Coding

```
#include<iostream>
using namespace std;
int main()
{
    int T;
    cin>>T;
    for(int j=1;j<=T;j++)
    {
        int N;
        cin>>N;
        int i,sum=0,count=0;
        i=1;
        while(sum<N)
        {
            sum=sum+i;
            i++;
            count++;
        }
        cout<<count<<endl;
    }
    return 0;
}
```

Fig 2.2.1

2.3 Output

```
2
10
4
5
3
-----
Process exited after 9.407 seconds with return value 0
Press any key to continue . . .
```

Fig 2.3.1

3 Problem 2

The Vita Sum

3.1 Problem Statement

Tom the cat is brushing up his Math skills. He has a bag containing N balls of different colors. Now Tom can randomly pick any even number of balls from the bag. Tom wants to find out the sum of all such combinations of balls that he can pull out from the bag. He can pull out at max K balls in one pick.

Input Format

First line contains two space separated numbers N and K

Output Format

The output is the sum of all the combinations of balls he can pull out modulo 10^9+7 i.e. (1000000007)

Constraints

- $0 \leq N, k \leq 10^{14}$
- $N \geq k$

Input

4

4

Output

8

Concepts Used To Solve:-

Long double : long double refers to a floating-point data type.

while loop : It is used to repeat a section of code an unknown number of times until a specific condition is met.

3.2 Coding

```
#include <stdio.h>

long double factorial( int t) {
    long double fact = 1;
    int i;
    for ( i = 1; i<= t; i++)
        fact = fact * i;
    return( fact);
}

int combination ( int n, int r) {
    return( factorial( n) / (factorial( r) * factorial(n- r) ) ) ;
}

int main() {
    int n , r;
    int i=0,vitaSum=0;
    scanf("%d%d",&n,&r);
    if(n>=r)
    {
        while( n>=i&&i<=r) {
            vitaSum+=combination(n,i);
            i+=2;
        }
    }
    printf("%d",vitaSum);
    return 0;
}
```

Fig 3.2.1

3.3 Output

```
4
4
8
-----
Process exited after 15.38 seconds with return value 0
Press any key to continue . . .
```

Fig 3.3.1

4 Problem 3

Reverse gear

4.1 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 Format

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 Format

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

Constraints

First move will always be in backward direction

$1 \leq N \leq 100$

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.

Concepts Used To Solve:-

While loop : It is used to repeat a section of code an unknown number of times until a specific condition is met.

Break Statement : which is used to bring the program control out of the loop.

Input

2
6 9 3 18
3 7 5 20

Output

162
220

4.2 Coding

```
#include<stdio.h>
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    for(int i=0;i<n;i++){
        int f,b,t,d,cnt=0,x;
        cin>>f>>b>>t>>d;
        while(1){
            if(b<d){
                x=((f+b)*t);
                cnt=cnt+x;
                d=d-(b-f);
            }
            else{
                x=(d*t);
                cnt=cnt+x;
                break;
            }
        }
        cout<<cnt<<endl;
    }
    return 0;
}
```

Fig 4.2.1

4.3 Output

```
2
6 9 3 18
162
3 7 5 20
220

-----
Process exited after 23.17 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.

Concepts Used To Solve:-

If - else : If the Boolean expression evaluates to true, then the if block will be executed, otherwise, the else block will be executed.

For loop : It is a control flow statement for specifying iteration, which allows code to be executed repeatedly.

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==0)
        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;
}
```

Fig 5.2.1

5.3 Output

```
20
2
-----
Process exited after 13.07 seconds with return value 0
Press any key to continue . . .
```

Fig 5.3.1

6. 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. Software Requirements

There are two ways to execute this projects

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

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 softwares are based on the platforms