**PROBLEM SOLVING**

(Solving Various Problems Using C Language & Python)

*Summer Internship Report Submitted in partial fulfilment*

*of the requirement for under graduate degree of*

**Bachelor of Technology**

In

**Computer Science Engineering**

By

**Vengalachetty Haneesha**

**221710307063**

<https://github.com/haneeshav04>

*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 Language and Python**” 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           VENGALACHETTY HANEESHA

Date: 20-07-2020                                                                                                      221710307063



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 Language and Python”** is being submitted by VENGALACHETTY HANEESHA (221710307063) 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.

                                                                                    VENGALACHETTY HANEESHA

                                                                                                                   221710307063

**TABLE OF CONTENTS**

[1 Introduction to the project](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.gjdgxs)  1

[2 Problem 1 – Longest Progressive Sequence](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.30j0zll)  2

[2.1 Problem Statement](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.1fob9te)  2

[2.2 Coding](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.3znysh7) 3

[2.3 Output](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.2et92p0) 3

[3 Problem 2 – Zombie World](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.tyjcwt)  4

[3.1 Problem Statement](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.3dy6vkm)  4

[3.2 Coding](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.1t3h5sf) 6

[3.3 Output](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.4d34og8) 7

[4 Problem 3 – Distribute Books](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.2s8eyo1)  8

[4.1 Problem Statement](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.17dp8vu)  8

[4.2 Coding](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.3rdcrjn)  9

[4.3 Output](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.26in1rg)  9

[5 Problem 4 – Prime Fibonacci](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.lnxbz9)  10

[5.1 Problem Statement](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.35nkun2)  10

[5.2 Coding](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.1ksv4uv) 12

[5.3 Output](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.44sinio)  14

[6 Problem 5 – Television Sets](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.z337ya)  15

[6.1 Problem Statement](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.3j2qqm3)  15

[6.2 Coding](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.1y810tw) 17

[6.3 Output](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.4i7ojhp)  19

[7 Problem 6](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.2xcytpi) – Collecting Candies 20

[7.1 Problem Statement](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.1ci93xb)  20

[7.2 Coding](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.3whwml4) 22

[7.3 Output](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.2bn6wsx)  24

[8 Software Requirements](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.ihv636) 25

[8.1 Hardware Requirements](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.32hioqz)  25

[8.2 Software Requirements](https://docs.google.com/document/d/1tjcjW_gBxLXJJ8lcM_zoOmIZgnojR5jVKL0I-zXZKt8/edit#heading=h.1hmsyys) 25

9 Bibliography 26

**1 Introduction to the project**

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

1. **Longest Progressive Sequence** - In this problem we have to find the longest progressive sequence in the given sequence of numbers.
2. **Zombie World** - In this problem we have to find out if Bob has enough energy to fight all the zombies in the game based on the given formula.
3. **Distribute Books** - In this problem we calculate the number of ways for the students to exchange the books so that they do not have the same book.
4. **Prime Fibonacci** - In this problem, in the given range find prime numbers and combine them to make 3-digit or 4-digit numbers. After combining get the prime numbers count and out of all the prime number find the largest and smallest prime numbers and compute Fibonacci series using the smallest and largest prime numbers.
5. **Television Sets** - In this problem we calculate the number of televisions to install to in the hospital to obtain the revenue target.
6. **Collecting Candies** - In this problem we should compute the minimum amount of time needed to collect all the candies in one box.

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

**2 Problem 2**

**Longest Progressive Sequence**

**2.1 Problem Statement**  
A sequence is said to be progressive if it doesn’t decrease at any point in time. For example, 1 1 2 2 is a progressive sequence but 1 2 1 is not a progressive sequence. Let S be the sequence and be represented by L spaced integers Ki, now your task is to find out the first longest progressive sequence present in the given sequence (S).

**Input Format:**  
First line will contain T, the length of the sequence and next line will contain T spaced integers Ki (where i = 0,1, …,L).  
Line 1 T, where T is the length of the sequence  
Line 2 Ki, where Ki is integer in sequence separated by space  
**Constraints:**  
1<=T<=10^6(one million)  
1<=Ki<=10^9(one billion)

**Output Format:**  
Line 1 longest progressive sequence present in the given sequence

**Sample Test Cases:**

**Input:**

4

1 1 2 1

**Output:**

1 1 2

**Input:**

5

1 2 1 2 2

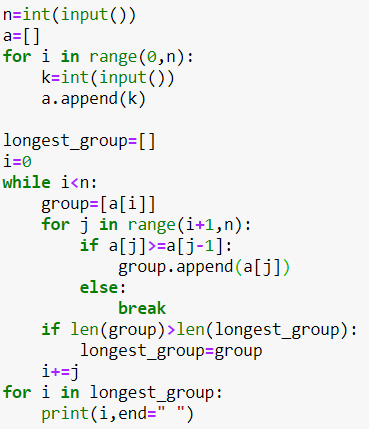
**Output:**

1 2 2

**Concepts Used:**

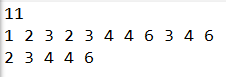
For loop with if condition nested is used to check List data structure was used to solve the problem to finally print longest progressive sequence.

**2.2 Coding**



**Fig 2.2.1**

**2.3 Output**



**Fig 2.3.1**

**3 Problem 2**

**Zombie World**

**3.1 Problem Statement**

One day Bob is playing Zombie World video game. In Zombie World game each round will contain N zombie's and each zombie's energy is Zi (where 1<=i<=N). Bob will start the round with B energy. In order to move to the next level Bob, need to kill all the N zombie's but Bob can select any one among N Zombie's. If energy of Bob (B) is less than Zombie energy (Zi) then Bob will die and lose the round else Bob will won, during the fighting with zombie, Bob will lose his energy by (Zi%2) + (Zi/2). At any point of game Bob will play optimally. Now your task is to find out whether Bob can reach to the next level or not.

**Input Format:**  
First line will contain B and N, separated by space, where B is the energy of Bob and N is the number of Zombie. Next line will contain N spaced integers each will represent the energy of zombie.  
Line 1 B N, where B is the energy of Bob and N is the number of Zombie  
Line 2 Zi, where Zi is a list containing energy of zombies separated by space  
**Constraints:**  
1<=N<=10^4  
1<=B<=10^9  
1<=Zi<=10^5  
Note:  
For this problem all the divisions are integer divisions.

**Output Format:**

For Valid Input:  
Print "YES" or "NO" depending upon whether Bob can reach the next level or not.

For Invalid Input:

Print “Invalid Input”

**Sample Test Cases:**

**Input:**  
35 3  
5 9 6

**Output:**  
YES

**Input:**  
4 4  
1 3 2 4

**Output:**  
NO

**Input:**

456 68

a

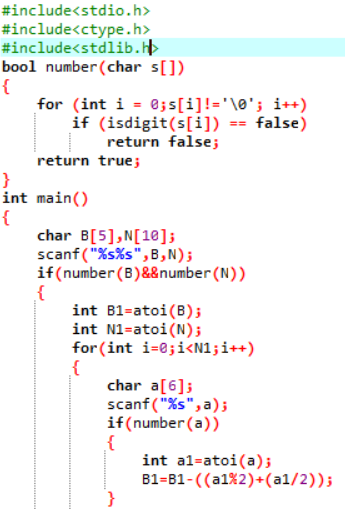
**Output:**

Invalid Input

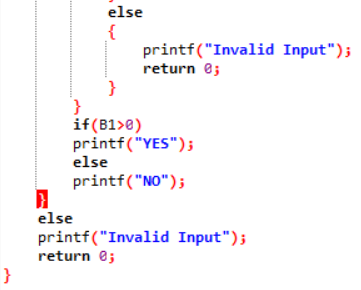
**Concepts Used:**

To check for the invalid input the function isdigit is used to check if giving input is integer or not. If condition is satisfied then proceed with the program otherwise print “Invalid Input” and terminate the program. If the program does not terminate calculate Bob’s energy based on the provided formula to check if can survive till the end of the game.

**3.2 Coding**



**Fig 3.2.1**

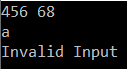


**Fig 3.2.2**

**3.3 Output**



**Fig 3.3.1**



**Fig 3.3.2**



**Fig 3.3.3**

**4 Problem 3**

**Distribute Books**

**4.1 Problem Statement**

For enhancing the book reading, school distributed story books to students as part of the Children’s day celebrations. To increase the reading habit, the class teacher decided to exchange the books every week so that everyone will have a different book to read. She wants to know how many possible exchanges are possible. If they have 4 books and students, the possible exchanges are 9. Bi is the book of i-th student and after the exchange he should get a different book, other than Bi.

B1 B2 B3 B4 – first state, before exchange of the books  
B2 B1 B4 B3  
B2 B3 B4 B1  
B2 B4 B1 B3  
B3 B1 B4 B2  
B3 B4 B1 B2  
B3 B4 B2 B1  
B4 B1 B2 B3  
B4 B3 B1 B2  
B4 B3 B2 B1

Find the number of possible exchanges, if the books are exchanged so that every student will receive a different book.

**Input Format:**

Input contains one line with N, indicates the number of books and number of students.

**Constraints:**

1<= N <= 1000000

**Output Format:**

Output the answer modulo 1000000007.

**Sample Test Cases:**

**Input:**

4

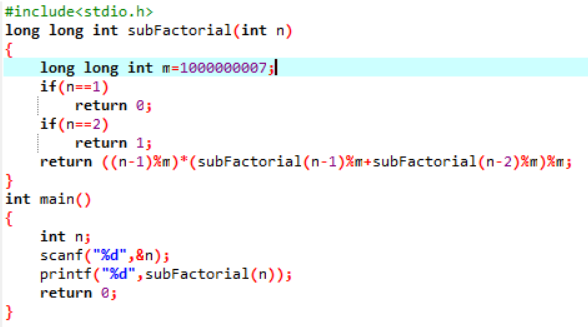
**Output:**

9

**Concepts Used:**

Recursion was used to obtain the sub factorial of the given number.

**4.2 Coding**



**Fig 4.2.1**

**4.3 Output**



**Fig 4.3.1**

**5 Problem 4**

**Prime Fibonacci**

**5.1 Problem Statement**

1. Given two numbers n1 and n2
2. Find prime numbers between n1 and n2, then
3. Make all possible unique combinations of numbers from the list of the prime numbers you found in step 1.
4. From this new list, again find all prime numbers.
5. Find the smallest (a) and largest (b) number from the 2nd generated list, also count of this list.
6. Consider the smallest and largest number as the 1st and 2nd number to generate the Fibonacci series respectively till the count (number of primes in the 2nd list).
7. Print the last number of a Fibonacci series as an output

**Input Format:**

One line containing two space-separated integers n1 and n2.

**Constraints:**

2 <= n1, n2 <= 100

n2 - n1 >= 35

**Output Format:**

The last number of a generated Fibonacci series.

**Sample Test Cases:**

**Input:**

2 40

**Output:**

13158006689

**Explanation:**

1st prime list = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]

Combination of all the primes = [23, 25, 27, 211, 213, 217, 219, 223, 229, 231, 32, 35, 37, 311, 313, 319, 323, 329, 331, 337, 52, 53, 57, 511, 513, 517, 519, 523, 529, 531, 537, 72, 73, 75, 711, 713, 717, 719, 723, 729, 731, 737, 112, 113, 115, 117, 1113, 1117, 1119, 1123, 1129, 1131, 1137, 132, 133, 135, 137, 1311, 1317, 1319, 1323, 1329, 1331, 1337, 172, 173, 175, 177, 1711, 1713, 1719, 1723, 1729, 1731, 1737, 192, 193, 195, 197, 1911, 1913, 1917, 1923, 1929, 1931, 1937, 232, 233, 235, 237, 2311, 2313, 2317, 2319, 2329, 2331, 2337, 292, 293, 295, 297, 2911, 2913, 2917, 2919, 2923, 2931, 2937, 312, 315, 317, 3111, 3113, 3117, 3119, 3123, 3129, 3137, 372, 373, 375, 377, 3711, 3713, 3717, 3719, 3723, 3729, 3731]

2nd prime list=[193, 3137, 197, 2311, 3719, 73, 137, 331, 523, 1931, 719, 337, 211, 23, 1117, 223, 1123, 229, 37, 293, 2917, 1319, 1129, 233, 173, 3119, 113, 53, 373, 311, 313, 1913, 1723, 317]

smallest (a) = 23

largest (b) = 3719

Therefore, the last number of a Fibonacci series i.e. 34th Fibonacci number in the series that has 23 and 3719 as the first 2 numbers is 13158006689

**Input:**

30 70

**Output:**

2027041

**Explanation:**

1st prime list=[31, 37, 41, 43, 47, 53, 59, 61, 67]

2nd prime list generated form combination of 1st prime list = [3137, 5953, 5347, 6761, 3761, 4337, 6737, 6131, 3767, 4759, 4153, 3167, 4159, 6143]

the smallest prime in 2nd list=3137

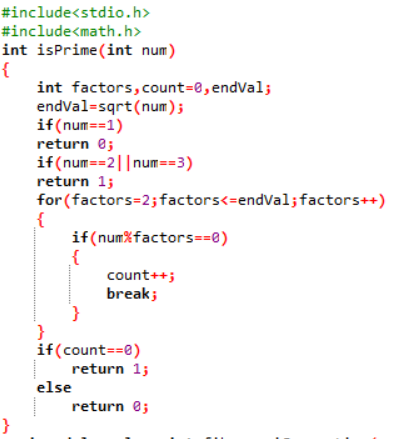
the largest prime in 2nd list=6761

Therefore, the last number of a Fibonacci series i.e. 14th Fibonacci number in the series that has 3137 and 6761 as the first 2 numbers is 2027041.

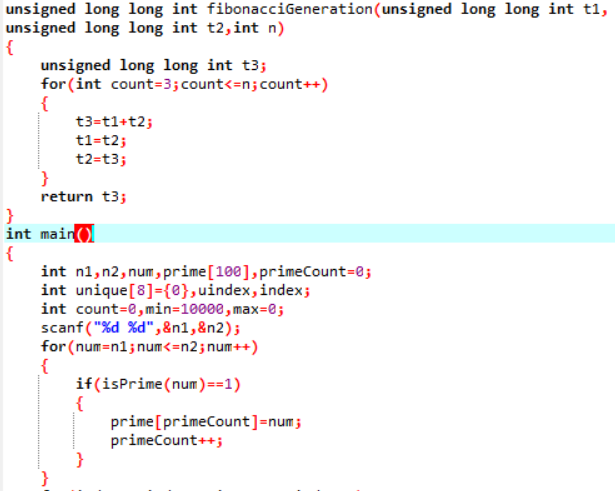
**Concepts Used:**

Basic C components like loops for obtaining the different combinations from the prime numbers and switch to check if a combination is repeating were used to execute this problem. There were multiple functions used like to check if given number is prime or not and to calculate the nth Fibonacci number.

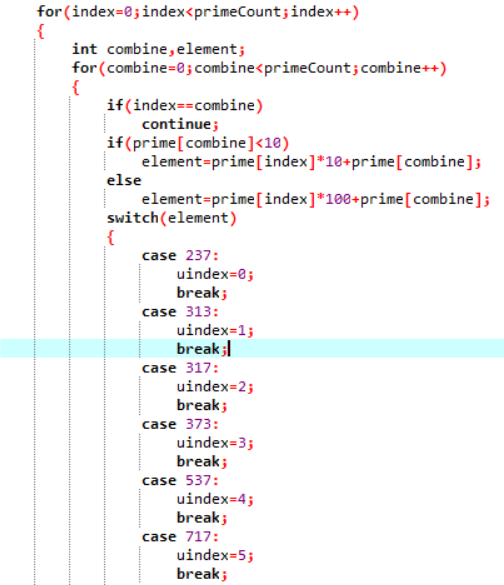
**5.2 Coding**



**Fig 5.2.1**



**Fig 5.2.1**



**Fig 5.2.3**



**Fig 5.2.4**

**5.3 Output**



**Fig 5.3.1**



**Fig 5.3.2**

**6 Problem 5**

**Television Sets**

**6.1 Problem Statement**

Dr. Vishnu is opening a new world class hospital in a small town designed to be the first preference of the patients in the city. Hospital has N rooms of two types – with TV and without TV, with daily rates of R1 and R2 respectively.

However, from his experience Dr. Vishnu knows that the number of patients is not constant throughout the year, instead it follows a pattern. The number of patients on any given day of the year is given by the following formula – (6-M)^2 + |D-15| ,

where M is the number of month (1 for jan, 2 for feb …12 for dec) and D is the date (1,2…31).

All patients prefer without TV rooms as they are cheaper, but will opt for with TV rooms only if without TV rooms are not available. Hospital has a revenue target for the first year of operation. Given this target and the values of N, R1 and R2 you need to identify the number of TVs the hospital should buy so that it meets the revenue target. Assume the Hospital opens on 1st Jan and year is a non-leap year.

**Input Format:**

First line provides an integer N that denotes the number of rooms in the hospital.

Second line provides two space-delimited integers that denote the rates of rooms with TV (R1) and without TV (R2) respectively.

Third line provides the revenue target.

**Constraints:**

Hospital opens on 1st Jan in an ordinary year

5 <= Number of rooms <= 100

500 <= Room Rates <= 5000

0 <= Target revenue < 90000000

**Output Format:**

Minimum number of TVs the hospital needs to buy to meet its revenue target. If it cannot achieve its target, print the total number of rooms in the hospital.

**Sample Test Cases:**

**Input:**

20

1500 1000

7000000

**Output:**

14  
**Explanation:**

Using the formula, number of patients on 1st Jan will be 39, on 2nd Jan will be 38 and so on. Considering there are only twenty rooms and rates of both type of rooms are 1500 and 1000 respectively, we will need 14 TV sets to get revenue of 7119500. With 13 TV sets Total revenue will be less than 7000000

**Input:**

10

1000 1500

10000000

**Output:**

10

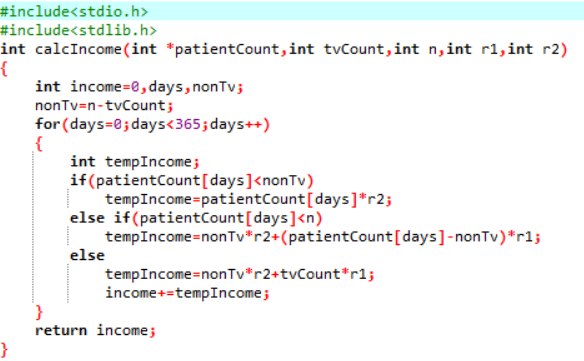
**Explanation:**

In the above example, the target will not be achieved, even by equipping all the rooms with TV. Hence, the answer is 10 i.e. total number of rooms in the hospital.

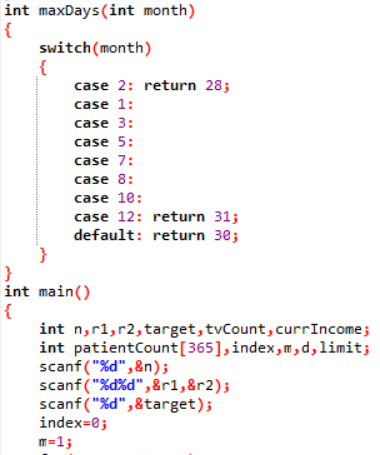
**Concepts Used:**

Basic components like switch, loops and functions were used to control the flow of execution along with binary search to obtain the number if televisions faster rather than linear search. Multiple functions were used in the program like to calculate the income for one year considering the different conditions that can affect the income calculation and a function to check the number of days in month.

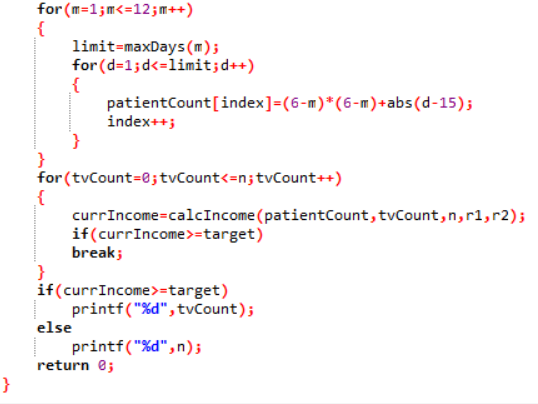
**6.2 Coding**



**Fig 6.2.1**

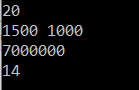


**Fig 6.2.2**

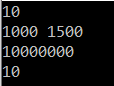


**Fig 6.2.3**

**6.3 Output**



**Fig 6.3.1**



**Fig 6.3.2**

**7 Problem 6**

**Collecting Candies**

**7.1 Problem Description**

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 Ci candies where Ci represents the total number of candies in the ith 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 an infinite number of empty boxes available.

At a time, he can pick up any two boxes for transferring and if both the boxes 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

**Constraints:**

1 < T < 10

1 < N< 10000

1 < [Candies in each box] < 100009

**Output Format:**

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

**Sample Test Cases:**

**Input:**

1

4

1 2 3 4

**Output:**

19

**Explanation:**

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.

**Input:**

1

5

1 2 3 4 5

**Output:**

33

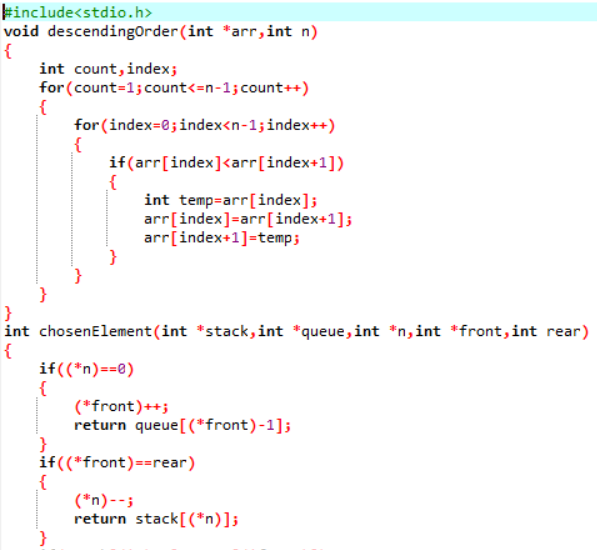
**Explanation:**

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 + 5 in a new box takes 9 seconds  
Adding 6 + 9 in a new box takes 15 seconds  
Hence total time taken is 33 seconds. There could be other combinations also, but overall time does not go below 33 seconds.

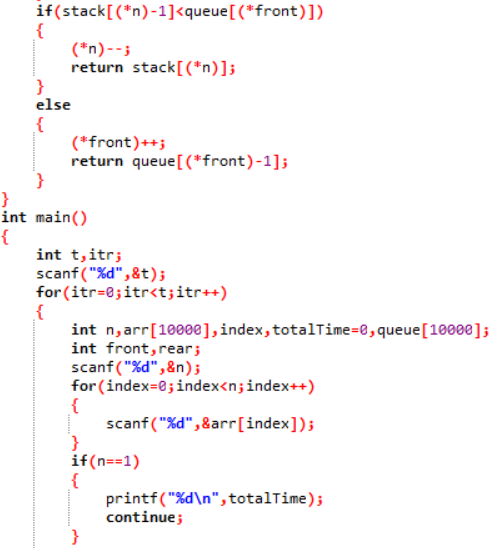
**Concepts Used:**

Data structures like stack and queue and bubble sort technique was used to combine the candies in the boxes to into one box in minimum amount of time. Multiple functions were used to perform specific task like sorting the boxes of candies in descending order and selecting a box of candies based on the number of candies in the box. This done till no boxes are left to combine which is done using a while loop.

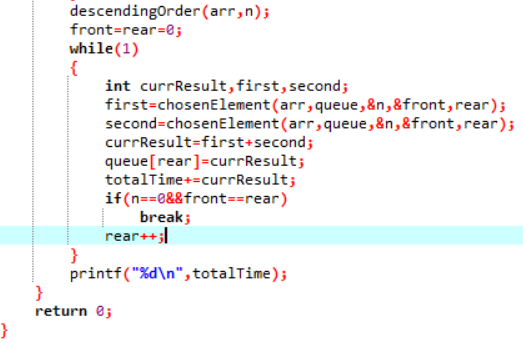
**7.2 Coding**



**Fig 7.2.1**



**Fig 7.2.2**

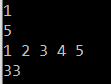


**Fig 7.2.3**

**7.3 Output**



**Fig 7.3.1**



**Fig 7.3.2**

# **10 Software Requirements**

## **10.1 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.

## **10.2 Software Requirements**

There are two ways to execute this project

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

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. This software is based on the platforms.

**BIBLIOGRAPHY**

1. <https://www.programminggeek.in/2016/08/CodeVita-2016-Round-2-Question-Collecting-Candies.html>
2. <https://www.programminggeek.in/2013/08/solution-of-sequence-problem-of-codevita-2013-organized-by-TCS.html#.Xx15OZ4zZPY>
3. <https://www.programmersdoor.com/post/tcs-codevita-prime-fibonacci#:~:text=Find%20the%20smallest%20(a)%20and,primes%20in%20the%202nd%20list).>
4. <https://codeofgeeks.com/television-sets/>
5. <https://www.programminggeek.in/2013/08/solution-of-zombie-world-problem-of-round1-of-codevita-2013-organized-by-TCS.html>
6. <https://codeofgeeks.com/distribute-books/>