KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)
(Approved by AICTE | Permanently Affiliated to JNTUK, Kakinada | Accredited by 'A' grade with NAAC | Accredited by NBA)
Vinjanampadu, Prathipadu Road, Vatticherukuru Mandal, Guntur – 522017 | www.kitsguntur.ac.in | 0863 – 2286666, 77, 88

Department of Computer Science and Engineering – Data Science

Course Structure – R20
(With effect from AY 2020-2021)

I Year - I Semester

S.No	Category	Course Code	Course Title	L	T	P	C
1	Engineering Science Course	20CS1T01	Problem Solving and Programming Using C	3	0	0	3
2	Basic Science Course	20SH1T04	Applied Chemistry	3	0	0	3
3	Basic Science Course	20SH1T06	Differential Equations	3	0	0	3
4	Engineering Science Course	20ME1T01	Engineering Graphics	1	0	4	3
5	Engineering Science Course	20EE1T02	Basics of Electrical andElectronics Engineering	3	0	0	3
6	Engineering Science Course (Lab)	20CS1L01	Problem Solving and Programming Using C Lab	0	0	3	1.5
7	Engineering Science Course (Lab)	20CS1L02	IT Workshop	0	0	3	1.5
8	Basic Science Course (Lab)	20SH1L04	Applied Chemistry Lab	0	0	3	1.5
			Total	**13**	**0**	**13**	**19.5**

Theory: BSC-2, ESC-3 **Practical:** BSC-1, ESC-2

Category	Credits
Basic Science Course	**7.5**
Engineering Science Course	**12**
Total	**19.5**

Department of Computer Science and Engineering – Data Science

I Year - II Semester

S.NO	Category	Course Code	Course Title	L	T	P	C
1	Humanities and Social Science Course	20SH2T01	Communicative English	3	0	0	3
2	Basic Science Course	20SH2T02	Applied Physics	3	0	0	3
3	Basic Science Course	20SH2T07	Linear Algebra & VectorCalculus	3	0	0	3
4	Engineering Science Course	20EC2T01	Digital Logic Design	3	0	0	3
5	Engineering Science Course	20CS2T01	Python Programming	3	0	0	3
6	Humanities and Social Science Course (LAB)	20SH2L01	Communicative English Skills Lab	0	0	3	1.5
7	Basic Science Course (LAB)	20SH2L02	Applied Physics Lab	0	0	3	1.5
8	Engineering Science Course (LAB)	20CS2L01	Python Programming Lab	0	0	3	1.5
9	Mandatory Course (AICTE suggested)	20GE2M01	Environmental Science	2	0	0	0
			Total	**17**	**0**	**9**	**19.5**

Theory: BSC-2, HSMC-1, ESC-2 **Practical:** BSC-1, HSMC-1, ESC-1 MC: 1

Category	Credits
Basic Science Course	**7.5**
Engineering Science Course	**7.5**
Humanities and Social Science	**4.5**
Mandatory Course (AICTE suggested)	**0**
Total	**19.5**

Department of Computer Science and Engineering – Data Science

Programme: Common to CSE,CAI,CSD,IT,ECE,EEE,ME, & CE		Semester: I			
Course Code	Course Name	L	T	P	C
20CS1T01	**PROBLEM SOLVING AND PROGRAMMING USING C**	3	0	0	3
Subject Category : ESC					

Course Objectives:
The objectives of Problem Solving and Programming Using C are

- To learn about the computer systems, computing environments, developing of a computer program and Structure of a C Program
- To gain knowledge of the operators, selection, control statements and repetition in C
- To learn about the design concepts of arrays, strings, enumerated structure and union types. To learn about their usage.
- To assimilate about pointers, dynamic memory allocation and know the significance of Preprocessor.
- To assimilate about File I/O and significance of functions.

Course Outcomes:
At the end of the Course, the student will be able to

CO 1: To write algorithms and to draw flowcharts for solving problems as well as compile and debug programs.

CO 2: To use different operators, data types and write programs that use two-way/ multi-way selection.

CO 3: To select the best loop construct for a given problem.

CO 4: To design and implement programs to analyze the different pointer applications.

CO 5: To decompose a problem into functions and to develop modular reusable code.

UNIT I: Introduction to Computers: Creating and running Programs, Computer Numbering System, Storing Integers, Storing Real Numbers Introduction to the C Language: Background, C Programs, Identifiers, Types, Variable, Constants, Input/output, Programming Examples, Type Qualifiers. Structure of a C Program: Expressions Precedence and Associativity, Side Effects, Evaluating Expressions, Type Conversion Statements, Simple Programs.

UNITII: Bitwise Operators: Exact Size Integer Types, Logical, Bitwise Operators, Shift Operators. Selection & Making Decisions: Logical Data and Operators, Two Way Selection, Multiway Selection, More Standard Functions. Repetition: Concept of Loop, Pretest and Post-test Loops, Initialization and Updating, Event and Counter Controlled Loops, Loops in C, Other Statements Related to Looping, Looping Applications, Programming Examples.

UNIT III: Arrays: Concepts, Using Array in C, Array Application, Two Dimensional Arrays, Multidimensional Arrays, Programming Example – Calculate Averages Strings: String Concepts, C String, String Input / Output Functions, Arrays of Strings, String Manipulation Functions String/ Data Conversion, A Programming Example – Morse Code Enumerated, Structure, and Union: The Type Definition (Type def), Enumerated Types, Structure, Unions, and Programming Application.

UNIT IV: Pointers: Introduction, Pointers to pointers, Compatibility, L value and R value Pointer Applications: Arrays, and Pointers, Pointer Arithmetic and Arrays, Memory Allocation Function, Array of Pointers, Programming Application. Processor Commands: Processor Commands.

UNIT V : Functions: Designing, Structured Programs, Function in C, User Defined Functions, Inter-Function Communication, Standard Functions, Storage Classes, Scope, life time, Passing Array to Functions, Passing Pointers to Functions, Command Line Arguments, Recursion Text Input / Output: Files, Streams, Standard Library Input / Output Functions, Formatting Input / Output Functions, Character Input / Output Functions Binary Input / Output: Text versus Binary Streams, Standard Library, Functions for Files, Converting File Type.

TEXT BOOKS:

1. Programming for Problem Solving, Behrouz A. Forouzan, Richard F.Gilberg, CENGAGE.
2. The C Programming Language, Brian W.Kernighan, Dennis M. Ritchie, 2e, Pearson.

REFERENCE BOOKS:

1. Computer Fundamentals and Programming, Sumithabha Das, McGraw Hill.
2. Programming in C, Ashok N. Kamthane, AmitKamthane, Pearson.
3. Computer Fundamentals and Programming in C, PradipDey, ManasGhosh, OXFORD.

E-RESOURCES:

1. https://www.tutorialspoint.com/cprogramming/index.htm
2. https://www.programiz.com/c-programming
3. https://www.javatpoint.com/c-programming-language-tutorial
4. https://nptel.ac.in/courses/106/104/106104128/

Department of Computer Science and Engineering – Data Science

Programme: Common to CSE,CAI,CSD,IT,ECE,EEE,ME, & CE					Semester: I
Course Code	Course Name	L	T	P	C
20CS1L01	**PROGRAMMING FOR PROBLEM SOLVING USING C LAB**	0	0	3	1.5
Subject Category : ESC					

Course Objectives:
The objectives of Programming for Problem Solving Using C Lab are

- Apply the principles of C language in problem solving.
- To design flowcharts, algorithms and knowing how to debug programs.
- To design & develop of C programs using arrays, strings pointers & functions.
- To review the file operations, preprocessor commands.

Course Outcomes:
At the end of the Course, the student will be able to

CO 1: Gains Knowledge on various concepts of a C language.
CO 2: Able to draw flowcharts and write algorithms.
CO 3: Able design and development of C problem solving skills.
CO 4: Able to design and develop modular programming skills.
CO 5: Able to trace and debug a program.

Exercise 1:
1. Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters.
2. Write a C program to compute the perimeter and area of a rectangle with a height of 7 inches and width of 5 inches.
3. Write a C program to display multiple variables.

Exercise 2:
1. Write a C program to calculate the distance between the two points.
2. Write a C program that accepts 4 integers p, q, r and s from the user where r and s are positive and p is even. If q is greater than r and s is greater than p and if the sum of r and s is greater than the sum of p and q print "Correct values", otherwise print "Wrong values".

Exercise 3:
1. Write a C program to convert a string to a long integer.
2. Write a program in C which is a Menu-Driven Program to compute the area of the various geometrical shape.
3. Write a C program to calculate the factorial of a given number.

Department of Computer Science and Engineering – Data Science

Exercise 4:
1. Write a program in C to display the n terms of even natural number and their sum.
2. Write a program in C to display the n terms of harmonic series and their sum. 1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms.
3. Write a C program to check whether a given number is an Armstrong number or not.

Exercise 5:
1. Write a program in C to print all unique elements in an array.
2. Write a program in C to separate odd and even integers in separate arrays.
3. Write a program in C to sort elements of array in ascending order.

Exercise 6:
1. Write a program in C for multiplication of two square Matrices.
2. Write a program in C to find transpose of a given matrix.

Exercise 7:
1. Write a program in C to search an element in a row wise and column wise sorted matrix.
2. Write a program in C to print individual characters of string in reverse order.

Exercise 8:
1. Write a program in C to compare two strings without using string library functions.
2. Write a program in C to copy one string to another string.

Exercise 9:
1. Write a C Program to Store Information Using Structures with Dynamically Memory Allocation
2. Write a program in C to demonstrate how to handle the pointers in the program.

Exercise 10:
1. Write a program in C to demonstrate the use of & (address of) and *(value at address) operator.
2. Write a program in C to add two numbers using pointers.

Exercise 11:
1. Write a program in C to add numbers using call by reference.
2. Write a program in C to find the largest element using Dynamic Memory Allocation.

Exercise 12:
1. Write a program in C to swap elements using call by reference.
2. Write a program in C to count the number of vowels and consonants in a string using a pointer.

Exercise 13:
1. Write a program in C to show how a function returning pointer.

2. Write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using malloc( ) function.

Exercise 14:
1. Write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using calloc( ) function. Understand the difference between the above two programs
2. Write a program in C to convert decimal number to binary number using the function.

Exercise 15:
1. Write a program in C to check whether a number is a prime number or not using the function.
2. Write a program in C to get the largest element of an array using the function.

Exercise 16:
1. Write a program in C to append multiple lines at the end of a text file.
2. Write a program in C to copy a file in another name.
3. Write a program in C to remove a file from the disk.

TEXT BOOKS:
1. Programming for Problem Solving, Behrouz A. Forouzan, Richard F.Gilberg, CENGAGE.
2. The C Programming Language, Brian W.Kernighan, Dennis M. Ritchie, 2e, Pearson.

REFERENCE BOOKS:
1. Computer Fundamentals and Programming, Sumithabha Das, McGraw Hill.
2. Programming in C, Ashok N. Kamthane, AmitKamthane, Pearson.
3. Computer Fundamentals and Programming in C, PradipDey, ManasGhosh, OXFORD.

E-RESOURCES:
1. https://www.tutorialspoint.com/cprogramming/index.htm
2. https://www.programiz.com/c-programming
3. https://www.javatpoint.com/c-programming-language-tutorial
4. https://nptel.ac.in/courses/106/104/106104128/

Department of Computer Science and Engineering – Data Science

Programme: Semester I: Common to CSE, CSD,CSI,ME & CE Semester: II: Common to ECE, EEE & IT					
Course Code	Course Name	L	T	P	C
20CS1L02	**IT Workshop**	0	0	3	1.5
Subject Category : ESC					

Course Objectives:
The objective of IT Workshop is to

- Explain the internal parts of a computer, peripherals, I/O ports, connecting cables.
- Demonstrate basic command line interface commands on Linux.
- Teach the usage of Internet for productivity and self paced lifelong learning.
- Demonstrate Office Tools such as Word processors, Spreadsheets and Presentation tools.

Course Outcomes:
At the end of the Course, the student will be able to

CO 1: Describe evolution of computers, storage devices, networking devices, transmission media and peripherals of a computer.
CO 2: Configure, evaluate and select hardware platforms for the implementation and execution of computer applications, services and systems
CO 3: Construct a fully functional virtual machine; summarize various Linux operating system commands.
CO 4: Develop presentation, documents and small applications using productivity tools such as word processor, presentation tools, spreadsheets, HTML, LaTex.

Computer Hardware:

Experiment 1:

Identification of peripherals of a PC, Laptop, Server and Smart Phones: Prepare a report containing the block diagram along with the configuration of each component and its functionality, Input/ Output devices, I/O ports and interfaces, main memory, cache memory and secondary storage technologies, digital storage basics, networking components and speeds.

Operating Systems:
Experiment 2:
Virtual Machine setup:

- Setting up and configuring a new Virtual Machine
- Setting up and configuring an existing Virtual Machine
- Exporting and packaging an existing Virtual Machine into a portable format

Department of Computer Science and Engineering – Data Science

Experiment 3:
Operating System installation:

- Installing an Operating System such as Linux on Computer hardware.

Experiment 4:
Linux Operating System Commands

- General command syntax ,Basic help commands, Basic File system commands, Date andtime.
- Basic Filters and Text processing, Basic File compression commands.
- Miscellaneous: apt-get, vi editor.

Experiment 5:
Networking Commands
ping, ssh, ifconfig, scp, netstat, ipstat, nslookup, traceroute, telnet, host, ftp, arp.

Internet Services:
Experiment 6:

- Web Browser usage and advanced settings like LAN, proxy, content, privacy, security, cookies, extensions/ plugins.
- Antivirus installation, configuring a firewall, blocking pop-ups .
- Email creation and usage, Creating a Digital Profile on LinkedIn.

Productivity Tools:&Office Tools
Experiment 7:
Demonstration and Practice on Text Editors like Notepad++, Sublime Text, Atom, Brackets, Visual code, etc

Experiment 8:
Demonstration and practice on Microsoft Word, Power Point.
Experiment 9:
Demonstration and practice on Microsoft Excel.

Experiment 10:
Demonstration and practice on LaTeX and produce professional PDF documents.

Experiment 11:
Internet of Things (IoT):
IoT fundamentals, applications, protocols, Architecture, IoT Devices communication models.

Introduction to HTML:
Experiment 12:
Understanding HTML tags and creation of simple web pages.
Assignment: Develop your home page using HTML Consisting of your photo, name, address and education details as a table and your skill set as a list.

Department of Computer Science and Engineering – Data Science

TEXT BOOKS:

1. Computer Fundamentals, Anita Goel, Pearson Education, 2017
2. PC Hardware Trouble Shooting Made Easy, TMH
3. Essential Computer and IT Fundamentals for Engineering and Science Students, Dr.N.B.Vekateswarlu, S.Chand.
4. HTML & CSS ,The Completee Reference , Fifth Edition , Thomas A. powell
5. LaTeX Companion – Leslie Lamport, PHI/Pearson.

REFERENCE BOOKS:

1. B. Govindarajulu, "IBM PC and Clones Hardware Trouble shooting and Maintenance", 2nd edition, Tata McGraw-Hill, 2002
2. "MOS study guide for word, Excel, Power point& Outlook Exams", Joan Lambert, JoyceCox, PHI.
3. "Introduction to Information Technology", ITL Education Solutions limited, Pearson Education.
4. Bigelows, "Trouble shooting, Maintaining& Repairing PCs", TMH.
5. Excel Functions and Formulas, Bernd held, Theodor Richardson, Third Edition

E-RESOURCES:

1. https://explorersposts.grc.nasa.gov/post631/20062007/computer_basics/ComputerPorts.doc
2. https://explorersposts.grc.nasa.gov/post631/2006-2007/bitsnbyte/Digital_Storage_Basics.doc
3. https://www.thegeekstuff.com/2009/07/linux-ls-command-examples
4. https://www.pcsuggest.com/basic-linux-commands/
5. https://www.vmware.com/pdf/VMwarePlayerManual10.pdf
6. https://gsuite.google.com/learning-center/products/#!/
7. https://www.raspberrypi.org

Department of Computer Science and Engineering – Data Science

Programme: Common to CSE,CAI,CSD,IT, &ECE					Semester:II
Course Code	Course Name	L	T	P	C
20CS2T01	**PYTHON PROGRAMMING**	3	0	0	3
Subject Category : ESC					

Course Objectives:
The Objectives of Python Programming are

- To learn about Python programming language syntax, semantics, and the runtime environment.
- To be familiarized with universal computer programming concepts like data types, containers
- To be familiarized with general computer programming concepts like conditional execution, loops &functions.
- To be familiarized with general coding techniques and object-oriented programming

Course Outcomes:
At the end of the Course, the student will be able to

CO1: Understand fundamentals of programming –variables, conditions, Lists, Tuples & Dictionaries.
CO2: Solve coding tasks related conditional execution, loops.
CO3: Impart Functions, Scope of variables, Modules, Packages.
CO4: Comprehend Concepts of File I/O, Exception Handling, Classes and Objects.
CO5: Develop general scientific programming through Matplotlib, NumPy and SciPy packages

UNIT I: Introduction to Python, Program Development Cycle, Input, Processing, and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations, Operators. Type conversions, Expressions, More about Data Output.
Data Types, and Expression: Strings Assignment, and Comment, Numeric Data Types and Character Sets.
Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables. **Programming:** Introduction to Programming Concepts with Scratch.

UNIT II: Repetition Structures: Introduction, while loop, for loop, Nested Loops. Control Statement: Definite iteration for Loop Formatting Text for output, Selection if and if else Statement Conditional Iteration The While Loop.
Strings and Text Files: Accessing Character and Substring in Strings, Data Encryption, Strings and Number Systems, String Methods TextFiles, string pattern matching. Understanding read functions, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Manipulating file pointer using seek, Programming using file operations.

Department of Computer Science and Engineering – Data Science

UNIT III: List and Dictionaries: Lists, tuple, Dictionaries and sets, frozen sets.
Design with Function: Defining Simple Functions, Functions as Abstraction Mechanisms, Problem Solving with Top Down Design, Design with Recursive Functions, Managing a Program's Namespace, Higher Order Function.
Modules: Modules, Standard Modules, NumPy, Pandas Packages.

UNIT IV: Error and Exceptions: Difference between an error and Exception, Detecting and Handling Exceptions, Raising Exceptions, Assertions, Built-in Exceptions, User Defined Exceptions**.**
Classes and Objects: Overview of OOP terminology, Creating Classes, Creating Instance Objects, Inheritance, Overriding Methods, Overloading Methods, Operators, Data hiding**.**

UNIT V: Simple plotting with pylab: Basic plotting, Labels, legends and customization, More advanced plotting
Matplotlib: Matplotlib basics, Contour plots, heatmaps and 3D plots.
NumPy: Basic array methods, Reading and writing an array to a file, Statistical methods, Polynomial, Linear algebra, Matrices, Random sampling, Discrete Fourier transforms**.**
SciPy: Physical constants and special functions, Integration and ordinary differential equations, Interpolation, Optimization, data-fitting and root-finding.

TEXT BOOKS:

1. Fundamentals of Python First Programs, Kenneth. A. Lambert,Cengage.
2. Learning Scientific Programming with Python, Christian Hill, Cambridge University Press.

REFERENCE BOOKS:

1. Introduction to Python Programming, Gowrishankar.S, VeenaA, CRC Press.
2. Introduction to Programming Using Python, Y. Daniel Liang,Pearson.
3. ReemaThareja, Python Programming using problem solving Approach, Oxford University Press 2017
4. R. NageswaraRao core python Programming second Edition.

E-RESOURCES:
1. https://www.tutorialspoint.com/python3/python_tutorial.pdf
2. https://bugs.python.org/file47781/Tutorial_EDIT.pdf
3. https://onlinecourses.nptel.ac.in/noc21_cs21/preview

Department of Computer Science and Engineering – Data Science

Programme: Common to CSE,CAI,CSD,IT, & ECE					Semester:II
Course Code	Course Name	L	T	P	C
20CS2L01	**PYTHON PROGRAMMING Lab**	0	0	3	1.5
Subject Category : ESC					

Course Objectives:
The Objectives of Python Programming Lab is

- To acquire programming skills in core Python.
- To acquire Object Oriented Skills in Python.
- To develop the skill of designing Graphical user Interfaces in Python.
- To develop the ability to write database applications in Python.

Course Outcomes:
At the end of the Course, the student will be able to

CO 1: Write, Test and Debug Python Programs
CO 2: Use Conditionals and Loops for Python Programs
CO 3: Use functions and represent Compound data using Lists, Tuples and Dictionaries
CO 4: Use various applications using python

Experiments

1) Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in akilogram.
2) Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total andaverage.
3) Write a program that uses a *for* loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86,89.
4) Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number oftimes.
5) Use a *for*loop to print a triangle like the one below. Allow the user to specify how high the triangle shouldbe.

```
*
**
***
****
```

6) Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right ornot.
7) Write a program that asks the user for two numbers and prints *Close* if the numbers are within .001 of each other and Not closeotherwise.
8) Write a program that asks the user to enter a word and prints out whether that word

contains anyvowels.

9) Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters *abcde*and*ABCDE* the program should print out*AaBbCcDdEe*

10) Write a program that asks the user for a large integer and inserts commas into it according to the standard American convention for commas in large numbers. For instance, if the user enters 1000000, the output should be1,000,000.

11) Write a program that generates a list of 20 random numbers between 1 and100.

(a) Print the list.

(b) Print the average of the elements in the list.

(c) Print the largest and smallest values in the list.

(d) Print the second largest and second smallest entries in the list

(e) Print how many even numbers are in the list.

12) Write a program to use split and join methods in the given string and store them in a dictionary data structure.

13) Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0].

14) Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you uselists.

15) Write a function called *sum_digits*that is given an integer num and returns the sum of the digits ofnum.

16) Write a function called *first_diff*thatis given two strings and returns the first location in which the strings differ. If the strings are identical, it should return-1.

17) Write a function called number_of_factorsthat takes an integer and returns how many factors the number has.

18) Write a function called *is_sorted*that is given a list and returns Trueif the list is sorted and False otherwise.

19) Write a function called root that is given a number x and an integer n and returns x1/n. In the function definition, set the default value of n to 2.

20) Write a function called merge that takes two already sorted lists of possibly different lengths, and merges them into a single sortedlist.

(a)Do this using the sort method. (b) Do this without using the sortmethod

21) Write a program that asks the user for a word and finds all the smaller words that can be made from the letters of that word. The number of occurrences of a letter in a smaller word can't exceed the number of occurrences of the letter in the user'sword.

Department of Computer Science and Engineering – Data Science

22) Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons.

23) Write a program that reads a list of temperatures from a file called *temps.txt*, converts those temperatures to Fahrenheit, and writes the results to a file calledftemps.txt.

24) Write a class called Product. The class should have fields called name, amount, and price, holding the product's name, the number of items of that product in stock, and the regular price of the product. There should be a method *get_price*that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called *make_purchase*that receives the number of items to be bought and decreases amount by that much.

25) Write a class called Time whose only field is a time in seconds. It should have a method called *convert_to_minutes*that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '5:50'. It should also have a method called *convert_to_hours*that returns a string of hours, minutes, and seconds formatted analogously to the previousmethod.

(a) Write a Python class to implement pow(x,n).

(b) Write a Python class to reverse a string word byword.

(c) Write a program to demonstrate Try/except/else.

(d) Write a function nearly _equal to test whether two strings are nearly equal. Two stringsa and b are nearly equal when a can be generated by a single mutation on b.

(e) Write a python program to create wheel using turtle graphics.

(f) Write a python program on GUI to create a Registration form.

(g) Write a python program to check whether a string starts and ends with the samecharacter or not (using Regular Expression re module).

26) ByteLand String :

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

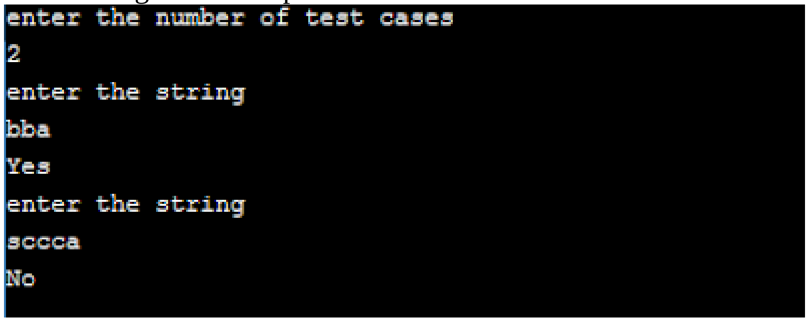
Output Format:
For each test case print "Yes" if the String "S" is super ascii, else print "No"

Constraints:
1<=T<=100
1<=|S|<=400, S will contains only lower case alphabets ('a'-'z').

Note: All text in bold which is corresponds to input and rest output.
Sample Input and Output :
2
bba

Yes
scca
No

Explanation:
In case 1. String "bba" -
The count of character 'b' is 2. Ascii value of 'b' is also 2.
The count of character 'a' is 1. Ascii value of 'a' is also 1.
Hence string "bba" is super ascii.



```
enter the number of test cases
2
enter the string
bba
Yes
enter the string
sccca
No
```

27) Given a sequence of distinct numbers a1, a2, ….. an, an inversion occurs if there are indices i<j such that ai > aj .
For example, in the sequence 2 1 4 3 there are 2 inversions (2 1) and (4 3).
The input will be a main sequence of N positive integers. From this sequence, a Derived Sequence will be obtained using the following rule. The output is the number of inversions in the derived sequence.

Rule for forming derived sequence

An integer may be represented in base 6 notation. In base 6, 10305 is 1x64 + 3x62 + 5 =1409. Note that none of the digits in that representation will be more than 5.
The sum of digits in a base 6 representation is the sum of all the digits at the various positions in the representation. Thus for the number 1409, the representation is 10305, and the sum of digits is 1+0+3+0+5=9. The sum of digits may be done in the decimal system, and does not need to be in base 6
The derived sequence is the sum of the digits when the corresponding integer is represented in the base 6 form number will be expressed in base 6, and the derived sequence is the sum of the digits of the number in the base 6 representation.

Constraints

N <= 50
Integers in sequence <= 107

Input Format

The first line of the input will have a single integer, which will give N.
The next line will consist of a comma separated string of N integers, which is the main sequence

Output

The number of inversions in the derived sequence formed from the main sequence.

Example 1

Input
5
55, 53, 88, 27, 33
Output
2

Explanation

The number of integers is 5, as specified in the first line. The given sequence is 55, 53, 88, 27, 33. The base 6 representation is 131, 125, 224, 43, 53 The derived sequence is 5,8,8,7,8 (corresponding to the sum of digits). The number of inversions in this is 2, namely (8, 7), (8, 7)

Example 2

Input

8
120,21,47,64,72,35,18,98

Output

11

Explanation

The base 6 representation of this is 320,33,115,144,200,55,30,242, and the derived sequence (sum of digits) is 5,6,7,9,2,10,3,8. The number of inversions is 11 (5,2), (5,3),(6,2) (6,3), (7,2), (7,3) (9,2),(9,3) (9,8),(10,3), (10,8)

TEXT BOOKS:

1. Fundamentals of Python First Programs, Kenneth. A. Lambert,Cengage.
2. Learning Scientific Programming with Python, Christian Hill, Cambridge University Press.

REFERENCE BOOKS:

1. Introduction to Python Programming, Gowrishankar.S, VeenaA, CRC Press.
2. Introduction to Programming Using Python, Y. Daniel Liang,Pearson.
3. ReemaThareja, Python Programming using problem solving Approach, Oxford University Press 2017
4. R. NageswaraRao core python Programming second Edition.

E-RESOURCES:

1. https://www.tutorialspoint.com/python3/python_tutorial.pdf
2. https://bugs.python.org/file47781/Tutorial_EDIT.pdf
3. https://onlinecourses.nptel.ac.in/noc21_cs21/preview