STES, Sinhgad academy of engineering, PuneDepartment

of Computer engineering

Laboratory manual (Instructor's Manual)

Course :210247

Data Structures Laboratory

Prepared by:

Mrs. M. A. Giri
Ms. M. M. Netke
Mrs. A. M. Todkar

Vision

उत्तमपुरुषान् उत्तमाभियंत्रृन् निर्मातुं कटीबध्दः वयम् !

"We are committed to produce not only good engineers but good human beings also."

Mission

"Holistic development of students and teachers is what we believe in and work for. We strive to achieve this by imbibing a unique value system, transparent work culture, excellent academic and physical environment conducive to learning, creativity and technology transfer. Our mandate is to generate, preserve and share knowledge for developing a vibrant society."

Department of Computer Engineering

Vision

“To build the Department as a Centre of Excellence for students in Computer Engineering.”

Mission

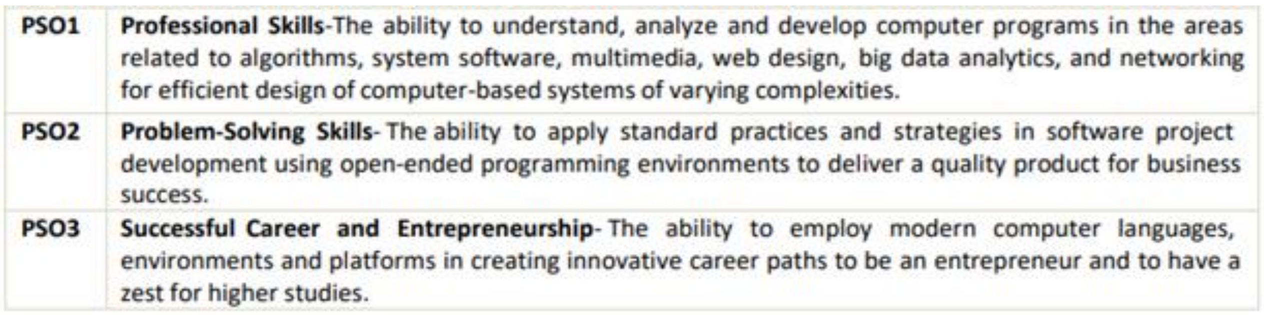
“We believe in developing value-based system for student and staff by providing healthy and transparent work culture to cultivate new ideas in the field of engineering and technology which will contribute to build a vibrant Society.”

Programme Outcomes (POs)

PO1	**Engineering knowledge**	Apply the knowledge of mathematics, science, Engineering fundamentals, and an Engineering specialization to the solution of complex Engineering problems.
PO2	**Problem analysis**	Identify, formulate, review research literature and analyze complex Engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and Engineering sciences.
PO3	**Design / Development of Solutions**	Design solutions for complex Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and Environmental considerations.
PO4	**Conduct Investigations of Complex Problems**	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern Tool Usage**	Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations.
PO6	**The Engineer and Society**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practices.
PO7	**Environment and Sustainability**	Understand the impact of the professional Engineering solutions in societal and Environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**	Apply ethical principles and commit to professional ethics and responsibilities and norms of Engineering practice.
PO9	**Individual and Team Work**	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication Skills**	Communicate effectively on complex Engineering activities with the Engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project Management and Finance**	Demonstrate knowledge and understanding of Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary Environments.
PO12	**Life-long Learning**	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs)

Computer Engineering graduate will be able to,



PSO1	**Professional Skills**-The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexities.
PSO2	**Problem-Solving Skills**- The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO3	**Successful Career and Entrepreneurship**- The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

Savitribai Phule Pune University
Second Year of Computer Engineering (2019 Course)
210246: Data Structures Laboratory

Teaching Scheme	Credit Scheme	Examination Scheme and Marks	
Practical: 04 Hours/Week	02	**Term Work:**	25 Marks
		Practical:	50 Marks

Companion Course : 210242: Fundamentals of Data Structures

Course Objectives:
To understand basic techniques and strategies of algorithm analysis, the memory requirement for various data structures like array, linked list, stack, queue etc using concepts of python and C++ programming language.

Course Outcomes:
On completion of the course, learner will be able to–

CO1: Use algorithms on various linear data structure using sequential organization to solve real life problems.
CO2: Analyze problems to **apply** suitable searching and sorting algorithm to various applications.
CO3: Analyze problems to **use variants of** linked list and solve various real life problems.
CO4: Designing and implement data structures and algorithms for solving different kinds of problems.

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface), University syllabus, conduction and Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work is done based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged. In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch **beyond the scope of syllabus.**

Set of suggested assignment list is provided in groups- A, B, C, D, and E. Each student must perform at least 13 assignments (at least 3 from group A, 3 from group B, 2 from group C, 2 from group D and 3 from group E.)

Group A and B assignments should be implemented in Python without using built-in methods for major functionality of assignment. Use List data structure of Python as array. Group C, D and E assignments should be implemented in C++ language.

Operating System recommended:- 64-bit Open source Linux or its derivative

Programming tools recommended: - Open Source Python, Programming tool like Jupyter Notebook, Pycharm, Spyder, G++/GCC.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Virtual Laboratory:

- http://cse01-iiith.vlabs.ac.in/Courses%20Aligned.html?domain=Computer%20Science

Suggested List of Laboratory Experiments/Assignments

Sr. No.	Group A
1	In second year computer engineering class, group A student's play cricket, group B students play badminton and group C students play football. Write a Python program using functions to compute following: - a) List of students who play both cricket and badminton b) List of students who play either cricket or badminton but not both c) Number of students who play neither cricket nor badminton d) Number of students who play cricket and football but not badminton. (Note- While realizing the group, duplicate entries should be avoided, Do not use SET built-in functions)
2	Write a Python program to store marks scored in subject "Fundamental of Data Structure" by N students in the class. Write functions to compute following: a) The average score of class b) Highest score and lowest score of class c) Count of students who were absent for the test d) Display mark with highest frequency
3	Write a **Python** program for department library which has N books, write functions for following: a) Delete the duplicate entries b) Display books in ascending order based on cost of books c) Count number of books with cost more than 500. d) Copy books in a new list which has cost less than 500.
4	Write a **Python** program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following: D 100 W 200 (Withdrawal is not allowed if balance is going negative. Write functions for withdraw and deposit) D means deposit while W means withdrawal. Suppose the following input is supplied to the program: D 300, D 300 , W 200, D 100 Then, the output should be: 500

5	Write a Python program to compute following operations on String: a) To display word with the longest length b) To determines the frequency of occurrence of particular character in the string c) To check whether given string is palindrome or not d) To display index of first appearance of the substring e) To count the occurrences of each word in a given string
6	It is decided that weekly greetings are to be furnished to wish the students having their birthdays in that week. The consolidated sorted list with desired categorical information is to be provided to the authority. Write a **Python** program to store students PRNs with date and month of birth. Let List_A and List_B be the two list for two SE Computer divisions. Lists are sorted on date and month. Merge these two lists into third list "List_SE_Comp_DOB" resulting in sorted information about Date of Birth of SE Computer students
7	Write a **Python** Program for magic square. A magic square is an n * n matrix of the integers 1 to n2 such that the sum of each row, column, and diagonal is the same. The figure given below is an example of magic square for case n=5. In this example, the common sum is 65.

15	**8**	**1**	**24**	**17**
16	**14**	**7**	**5**	**23**
22	**20**	**13**	**6**	**4**
3	**21**	**19**	**12**	**10**
9	**2**	**25**	**18**	**11**

8	Write a **Python** program that determines the location of a saddle point of matrix if one exists. An m x n matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row i and the largest value in j.
9	Write a **Python** program to compute following computation on matrix: a) Addition of two matrices B) Subtraction of two matrices c) Multiplication of two matrices d) Transpose of a matrix
10	Write a **Python** program for sparse matrix realization and operations on it- Transpose, Fast Transpose and addition of two matrices
	Group B
11	a) Write a **Python** program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search. b) Write a **Python** program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search
12	a) Write a **Python** program to store names and mobile numbers of your friends in sorted order on names. Search your friend from list using binary search (recursive and non-recursive). Insert friend if not present in phonebook b) Write a **Python** program to store names and mobile numbers of your friends in sorted order on names. Search your friend from list using Fibonacci search. Insert friend if not present in phonebook.
13	Write a **Python** program to maintain club members, sort on roll numbers in ascending order. Write function "Ternary_Search" to search whether particular student is member of club or not. Ternary search is modified binary search that divides array into 3 halves instead of two.
14	Write a **Python** program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a) Selection Sort b) Bubble sort and display top five scores.

15	Write a **Python** program to store second year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a) Insertion sort b) Shell Sort and display top five scores
16	Write a **Python** program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.
17	Write a **Python** program to store 12th class percentage of students in array. Write function for sorting array of floating point numbers in ascending order using bucket sort and display top five scores.
18	Write **Python** program to store 10th class percentage of students in array. Write function for sorting array of floating point numbers in ascending order using radix sort and display top five scores
	Group C
19	Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club member's information using singly linked list. Store student PRN and Name. Write functions to: a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) Two linked lists exists for two divisions. Concatenate two lists.
20	The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 10 rows and 7 seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with. Use array to store pointers (Head pointer) to each row. On demand a) The list of available seats is to be displayed b) The seats are to be booked c) The booking can be cancelled.
21	Write C++ program for storing appointment schedule for day. Appointments are booked randomly using linked list. Set start and end time and min and max duration for visit slot. Write functions for- A) Display free slots B) Book appointment C) Sort list based on time D) Cancel appointment (check validity, time bounds, availability) E) Sort list based on time using pointer manipulation
22	Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set B of students like butterscotch ice-cream. Write C++ program to store two sets using linked list. compute and display- a) Set of students who like both vanilla and butterscotch b) Set of students who like either vanilla or butterscotch or not both c) Number of students who like neither vanilla nor butterscotch
23	Write C++ program for storing binary number using doubly linked lists. Write functions- a) To compute 1's and 2's complement b) Add two binary numbers
24	Write C++ program to realize Set using Generalized Liked List (GLL) e.g. A ={ a, b, {c, d,e, {}, {f,g}, h, I, {j,k}, l, m}. Store and print as set notation.
	Group D

25	A palindrome is a string of character that's the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions- a) To print original string followed by reversed string using stack b) To check whether given string is palindrome or not
26	In any language program mostly syntax error occurs due to unbalancing delimiter such as (),{},[]. Write C++ program using stack to check whether given expression is well parenthesized or not.
27	Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions: 1. Operands and operator, both must be single character. 2. Input Postfix expression must be in a desired format. 3. Only '+', '-', '*' and '/ ' operators are expected.
28	A classic problem that can be solved by backtracking is called the Eight Queens problem, which comes from the game of chess. The chess board consists of 64 square arranged in an 8 by 8 grid. The board normally alternates between black and white square, but this is not relevant for the present problem. The queen can move as far as she wants in any direction, as long as she follows a straight line, Vertically, horizontally, or diagonally. Write C++ program with recursive function for generating all possible configurations for 4-queen's problem.
	Group E
29	Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue.
30	Write program to implement a priority queue in C++ using an inorder list to store the items in the queue. Create a class that includes the data items (which should be template) and the priority (which should be int). The inorder list should contain these objects, with operator <= overloaded so that the items with highest priority appear at the start of the list (which will make it relatively easy to retrieve the highest item.)
31	A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque.
32	Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array.

ASSIGNMENT LIST

Assignment No.	Name of the Assignment /Experiment
1	In second year computer engineering class, group A student's play cricket, group Bstudents play badminton and group C students play football. Write a P**ython** program using functions to compute following: - a) List of students who play both cricket and badminton b) List of students who play either cricket or badminton but not both c) Number of students who play neither cricket nor badminton d) Number of students who play cricket and football but not badminton. (Note- While realizing the group, duplicate entries should be avoided, Do not use SET built-in functions)
2	Write a Python program to store marks scored in subject "Fundamental of Data Structure" by N students in the class. Write functions to compute following: a) The average score of class b) Highest score and lowest score of class c) Count of students who were absent for the test d) Display mark with highest frequency
3	Write a **python** program to compute following computation on matrix: a) Addition of two matrices b) Subtraction of two matrices c) Multiplication of two matrices d) Transpose of a matrix
4	a) Write a **python** program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search. b) Write a **python** program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search.
5	Write a python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a) Selection Sort b) Bubble sort and display top five scores.
6	Write a **python** program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.
7	Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request.

	Similarly one may cancel the membership of club. First node is reserved for president of Club and last node is reserved for secretary of club. Write C++ program to maintain club member's information using singly linked list. Store student PRN and Name. Write functions to: a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) Two linked lists exist for two divisions. Concatenate two lists.
8	Second year Computer Engineering class, set A of students like Vanilla Ice-cream andset B of students like butterscotch ice-cream. Write C++ program to store two sets using linked list. compute and display- a) Set of students who like both vanilla and butterscotch b) Set of students who like either vanilla or butterscotch or not both c) Number of students who like neither vanilla nor butterscotch
9	A palindrome is a string of character that's the same forward and backward. Typically, Punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions- a) To print original string followed by reversed string using stack b) To check whether given string is palindrome or not
10	Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions: 1. Operands and operator, both must be single character. 2. Input Postfix expression must be in a desired format. 3. Only '+', '-', '*' and '/ ' operators are expected.
11	Queues are frequently used in computer programming, and a typical example is the Creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue.
12	A double-ended queue (deque) is a linear list in which additions and deletions may bemade at either end. Obtain a data representation mapping a deque into a one- dimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque
13	Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array.

Course Objectives

1. To understand basic techniques and strategies of algorithm analysis, the memory requirement for various data structures like array, linked list, stack, queue etc using concepts of python and C++ programming language.
2. Effective use of Object Oriented Programming concepts in data structures and files in programming.
3. Effective use of multi-core programming architecture in programming.
4. Effective use of latest programming tools like Eclipse Programming Framework, Microsoft Visual Studio, TC++, QT/ Java/Python etc.

Mapping Of Subjects With Cos and Pos

Course Name	Course Outcome	a	b	c	d	e	f	g	h	i	j	k	l
SE- 2019 Data Structure Lab	1. Effective use of Object Oriented Programming concepts in data structures and files in programming.	2	2	2	1						1		1
	2. Effective use of multi-core programming architecture in programming.	2	1	2	1		2				1		1
	3. Effective use of latest programming tools like Eclipse Programming Framework, MicrosoftVisual Studio, TC++, QT/ Java/Python etc	2	2	2	2	2	1			1	2		2

Mapping Of Assignments to Cos and Pos

Sr No	Name of the Assignment /Experiment	Course Outcomes achieved	PEO,s achieved
1	In second year computer engineering class, group A student's play cricket, group B students play badminton and group C students play football. Write a P**ython** program using functions to compute following: - a) List of students who play both cricket and badminton b) List of students who play either cricket or badminton but not both c) Number of students who play neither cricket nor badminton d) Number of students who play cricket and football but not badminton. (Note- While realizing the group, duplicate entries should be avoided, Do not use SET built-in functions)	1,2,3	a,b,c,d,e
2	Write a Python program to store marks scored in subject "Fundamentalof Data Structure" by N students in the class. Write functions to compute following: a) The average score of class b) Highest score and lowest score of class c) Count of students who were absent for the test d) Display mark with highest frequency	1,3	b,c,d,e
3	Write a **python** program to compute following computation onmatrix: a) Addition of two matrices b) Subtraction of two matrices c) Multiplication of two matrices d) Transpose of a matrix	1,2,3	a,b,c,d,e
4	a) Write a **python** program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search. b) Write a **python** program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search	1,3	b,c,d,e,j
5	Write a python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a) Selection Sort b) Bubble sort and display top five scores.	1,3	b,c,d,e,j

6	Write a **python** program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.	1,2,3	b,c,d,e
7	Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club members information using singly linked list. Store student PRN and Name. Write functions to: a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) Two linked lists exist for two divisions. Concatenate two lists.	1,3	a,b,c,d,e
8	Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set B of students like butterscotch ice-cream.Write C++ program to store two sets using linked list. compute anddisplay- a) Set of students who like both vanilla and butterscotch b) Set of students who like either vanilla or butterscotch or not both c) Number of students who like neither vanilla nor butterscotch.	1,3	a,b,c,d,e
9	A palindrome is a string of character that's the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions- a) To print original string followed by reversed string using stack b) To check whether given string is palindrome or not.	1,3	b,c,d,e
10	Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions: 1. Operands and operator, both must be single character. 2. Input Postfix expression must be in a desired format. 3. Only '+', '-', '*' and '/ ' operators are expected.	1,3	a,b,c,d,e
11	Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue.	1,3	a,b,c,d,e

12	A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque.	1,2,3	a,b,c,d,e
13	Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array.	1,2,3	a,b,c,d,e,j

Assignment No. 1

Problem Statement:

In second year computer engineering class, group A student’s play cricket, group B students play badminton and group C students play football.
Write a P**ython** program using functions to compute following: -
a) List of students who play both cricket and badminton
b) List of students who play either cricket or badminton but not both
c) Number of students who play neither cricket nor badminton
d) Number of students who play cricket and football but notbadminton.

Theory

Set-:

Set is a collection of elements.

The term class is used to denote set.

A set may contain finite or infinite no. of elements.

Empty set-: a set is called as empty set if it contains no element denoted by ϕ.

Universal set-: if all the set, subset of given set then this set is called as universal set denoted by U.Subset-:

if every element of set A is an element of set B, then we say that A subset of B.

Equality-> Two sets are said to be equal if they are subset of each other.

Operations of set-:

1. Union
2. Intersectionϕ
3. Difference
4. Complement
5. Cartesian product

1. Union- union of 2 sets A&B is the set containing all elements which are in A or B or in bothsets.
   e.g. A={1,2,3} and B={4,2,5} AUB={1,2,3,4,5}

2. Intersection-> it is the common elements between set A and Set B.
   e.g $A \cap B=\{2\}$
3. Difference ->
   Let A and B be 2 sets then A-B is defined asA-B={x/x € A and x € B}

E.g. A-B={1,3}

4. Complement- set containing all elements of universal set excepting its own elements.e.g. A={1,2,3}
   A_c= U-{1,2,3}
5. Cartesian product- it is the set whose members are all +ve ordered pair(a,b), where a ismember and b is member of B.

e.g.- a={a,b,c}

b={a,b} aXb={(a,a),(a,b),(b,a),(b,b),(c,a),(c,b)}**Algebra of sets-:**

1. Associative law-

(a) AUB=AU(BUC)
(b) $A \cap B \cap C=A \cap (B \cap C)$

2. Commutative laws-:

(a) AUB=BUA(b)$A \cap B=B \cap A$

3. Distributive law

(a) $AU(B \cap C)=(AUB)$[illegible] UC)

4. Independent laws-:

(a) AUB=A

(b) A∩ A=A

5. Identity laws-:

(a) AU∅=A

(b) A∩ U=A

(c) AUU=U

(d) A∩ ∅=∅

6. Complement laws-:

(a) $AUA_C=U$

(b) $A\cap A_c=\emptyset$

7. Absorption laws-:

(a) AU(A∩ B)=A

(b) A∩ (AU∅)=A8.Involution law-:

$A_c=A$

9. Demorgans law-:

(a) $(AUB)_c=AcUB_c$(b)$(A\ V)_c=AcUB_c$

∩

Conclusion:

we implemented set assignment successfully.

Assignment No. 2

Problem Statement:

Write a Python program to store marks scored in subject "Fundamentalof Data Structure" by N students in the class. Write functions to compute following:
a) The average score of class
b) Highest score and lowest score of class
c) Count of students who were absent for the test
d) Display mark with highest frequency

Introduction:

Algorithm is a sequence of unambiguous instruction for solving a problem.It is procedure of formula for solving a problem.
It is written in simple English.
It should be unique and self-explanatory.

Characteristic

- Precision: Steps are precisely defined.
- Uniqueness: Results of each folder are uniquely defined.
- Finiteness: Algorithm stops after finite number of steps.
- Input: Algorithm receives input.
- Output: Algorithm produces proper output.
- Generality: Algorithm applies is set input.

Complexity of an Algorithm

Complexity of an algorithm is a measure of amount of time and space required by it. For input ofgiven size „n" we define complexity as a numerical function. T(n)- Time vs. input size.

Time Complexities
It is total time required by algorithm to run for completion. It is expressed in big0 notation.

Space Complexities
Amount of computer memory required during program execution as a function of input size. Thereare 3 scenarios:

i. Best case: Least amount of time.

ii. Worst Case: Maximum amount of time.

iii. Average Case: Average amount of time.

Introduction to Data Structure
Data structure is a specified format for organizing and storing data. It includes array, files, records,tables, trees and so on.

Types of Data Structure

1. Linear
   Elements are organized sequentially.

2. Non Linear
   Data Elements can be attached to several other elements. Example- Tree, graph.

3. Static
   The size of structure is fixed. Storing and performing operations is easy.

4. Dynamic
   Size of structure is not defined. Difficult for storing or performing operation. Memory isallocated at execution time.

Conclusion:

Hence, we have studied the concept of data structure and algorithm.

Assignment No. 3

Problem Statement:

Write a **python** program to compute following computation on matrix:
a) Addition of two matrices
b) Subtraction of two matrices
c) Multiplication of two matrices
d) Transpose of a matrix

Introduction:

Introduction to Array

Array is a data structure that can store fixed size sequential collection of element of same type. An array is used for storing a collection of data. But it is often more useful to think of an array ascollection of variables of same types.

Instead of declaring individual variables you declare one array variable and use number [0]. Aspecific element in array is accessed by index.

All arrays consists of contiguous memory locations the lower address corresponds to firstelement and highest address to last element.

num[0]	num[1]	num[2]…………………	last element
First element	Second Element	Third Element	

Single dimensional array

The one dimensional array in C + + is simplest type of array that contains only one row hassingle set of square „[ ]‟ bracket.

eg: int age[ ]=New int[6]

The array of age is one dimensional containing only 6 elements in a single row

Two dimensional array

An array track of multiple pieces of information in linear order is a 1D array. A 2D array isnothing more than an array of arrays.

A local 1D array looks likeNew int [6]

And 2D array looks like

```
int[ ] [ ] my array={(0, 1, 2, 3) (3, 2, 1, 0) (3, 5, 6, 1) (3, 8, 3, 4)}
```

For our purpose it is better to think of 2D array as a matrix.

Illustration

```
int[ ] my array={(0, 1, 2, 3)
                 (3, 2, 1, 0)
                 (3, 5, 6, 1)
                 (3, 8, 3, 4)}
```

Multidimensional Array

Array having more than subscript variable is called multidimensional array.Array with more than 2 square brackets is also called a matrix.

int[ ][ ][ ] my array is an example of multidimensional array

Memory Representation and Address Calculation

1. 2D arrays are stored in continuous memory location row wise. stored in continuous
2. 3x3 array is shown below in first diagram.
3. Consider 3x3 array is stored which starts from 4000.
4. Array element [0] will be stored at 4000 again a[0][1] will be stored in next memory.
5. After element of first row and stored appropriate memory location element of next rowgets their corresponding memory locations.

	col 0	col 1	col 2
Row 0	1	2	3
Row 1	4	5	6
Row 2	7	8	9

Element	Mem location
a[0][0]	4000
a[0][1]	4002
a[0][2]	4004
a[1][0]	4006
a[1][1]	4008
a[1][2]	4010
a[2][0]	4012
a[2][1]	4014
a[2][2]	4016

1	2	3
4000	4002	4004
4	5	6
4006	4008	4010
7	8	9
4012	4014	4016

Matrix Operations

1. Addition

$$\begin{matrix} 2 & 3 \\ 2 & 4 \end{matrix} + \begin{matrix} 1 & 4 \\ 5 & 6 \end{matrix} = \begin{matrix} 3 & 7 \\ 7 & 10 \end{matrix}$$

2. Multiplication

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \text{ X } \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} = \begin{matrix} 7 & 10 \\ 15 & 22 \end{matrix}$$

3. Transpose

$$A= \begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix} \quad A\char`^T= \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

4. Subtraction

$$\begin{matrix}2 & 3\\ 4 & 5\end{matrix} \quad - \quad \begin{matrix}1 & 2\\ 3 & 3\end{matrix} \quad = \quad \begin{matrix}1 & 1\\ 1 & 2\end{matrix}$$

Conclusion:

Hence, we have learned to use array.

Assignment No. 4

Problem Statement:

a) Write a python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search.

b) Write a python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search

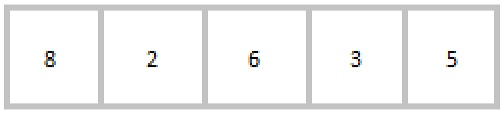
Introduction:

Linear Search

A linear search is the basic and simple search algorithm. A linear search searches an element or value from an array till the desired element or value is not found and it searches in a sequence order. It compares the element with all the other elements given in the list and if the element is matched it returns the value index else it return -1. Linear Search is applied on the unsorted or unordered list when there are fewer elements in a list.

Example with Implementation

To search the element 5 it will go step by step in a sequence order.



```
function findIndex(values, target)
 {
   for(var i = 0; i < values.length; ++i)
```

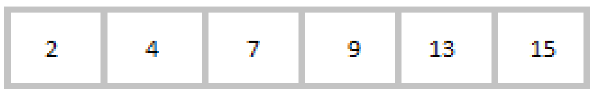
```
    {
      if (values[i] == target)
        {
          return i;
        }
    }
  return -1;
 }
//call the function findIndex with array and number to be searched
findIndex([ 8 , 2 , 6 , 3 , 5 ] , 5) ;
```

Binary Search

Binary Search is applied on the sorted array or list. In binary search, we first compare the value with the elements in the middle position of the array. If the value is matched, then we return the value. If the value is less than the middle element, then it must lie in the lower half of the array and if it's greater than the element then it must lie in the upper half of the array. We repeat this procedure on the lower (or upper) half of the array. Binary Search is useful when there are large numbers of elements in an array.

Example with Implementation

To search an element 13 from the sorted array or list.



- As we can see the above array is sorted in ascending order.
- Binary Search is applied on sorted lists only, so that we can make the search fast, by breaking the list everytime.
- Start with middle element,
- if its EQUAL to the number we are searching, then RETURN
- if its less than it, then move to the RIGHT.
- if its more that it, then move to the LEFT.
- And then, REPEAT, till you find the number.

```
function  findIndex(values, target)
{
  return binarySearch(values, target, 0, values.length - 1);
};

function binarySearch(values, target, start, end) {
  if (start > end) { return -1; } //does not exist

  var middle = Math.floor((start + end) / 2);
  var value = values[middle];

  if (value > target) { return binarySearch(values, target, start, middle-1); }
  if (value < target) { return binarySearch(values, target, middle+1, end); }
  return middle; //found!
}
findIndex([2, 4, 7, 9, 13, 15], 13);
```

In the above program logic, we are first comparing the middle number of the list, with the target, if it matcheswe return. If it doesn't, we see whether the middle number is greater than or smaller than the target.

If the Middle number is greater than the Target, we start the binary search again, but this time on the left halfof the list, that is from the start of the list to the middle, not beyond that.

If the Middle number is smaller than the Target, we start the binary search again, but on the right half of thelist, that is from the middle of the list to the end of the list.

Conclusion:

Hence we studied searching & sorting.

Assignment No. 5

Problem Statement:

Write a python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using
a) Selection Sort
b) Bubble sort and display top five scores.

Introduction:

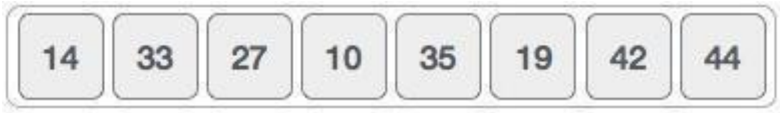
Selection Sort

Selection sort is a simple sorting algorithm. This sorting algorithm is a in-place comparison based algorithm in which the list is divided into two parts, sorted part at left end and unsorted part at right end. Initially sorted part is empty and unsorted part is entire list.

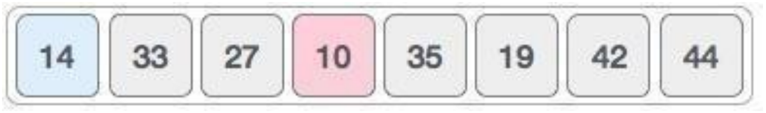
Smallest element is selected from the unsorted array and swapped with the leftmost element and that element becomes part of sorted array. This process continues moving unsorted array boundary by one element to the right.

This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n^2)where n are no. of items.

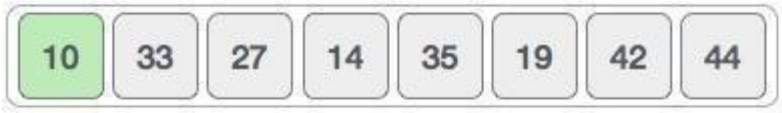
Working of selection Sort:



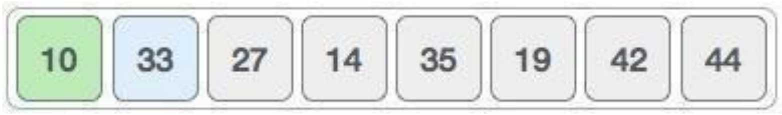
For the first position in the sorted list, the whole list is scanned sequentially. The first position where14 is stored presently, we search the whole list and find that 10 is the lowest value.



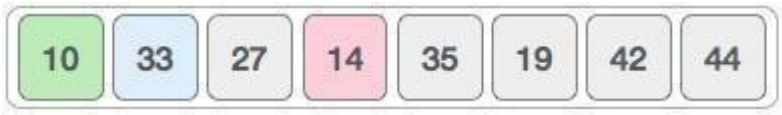
So we replace 14 with 10. After one iteration 10, which happens to be the minimum value in the list, appears in the first position of sorted list.



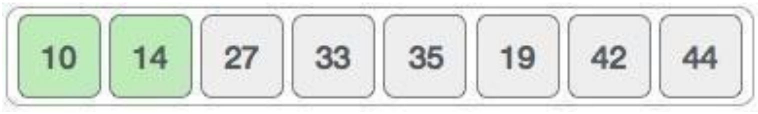
For the second position, where 33 is residing, we start scanning the rest of the list in linear manner.



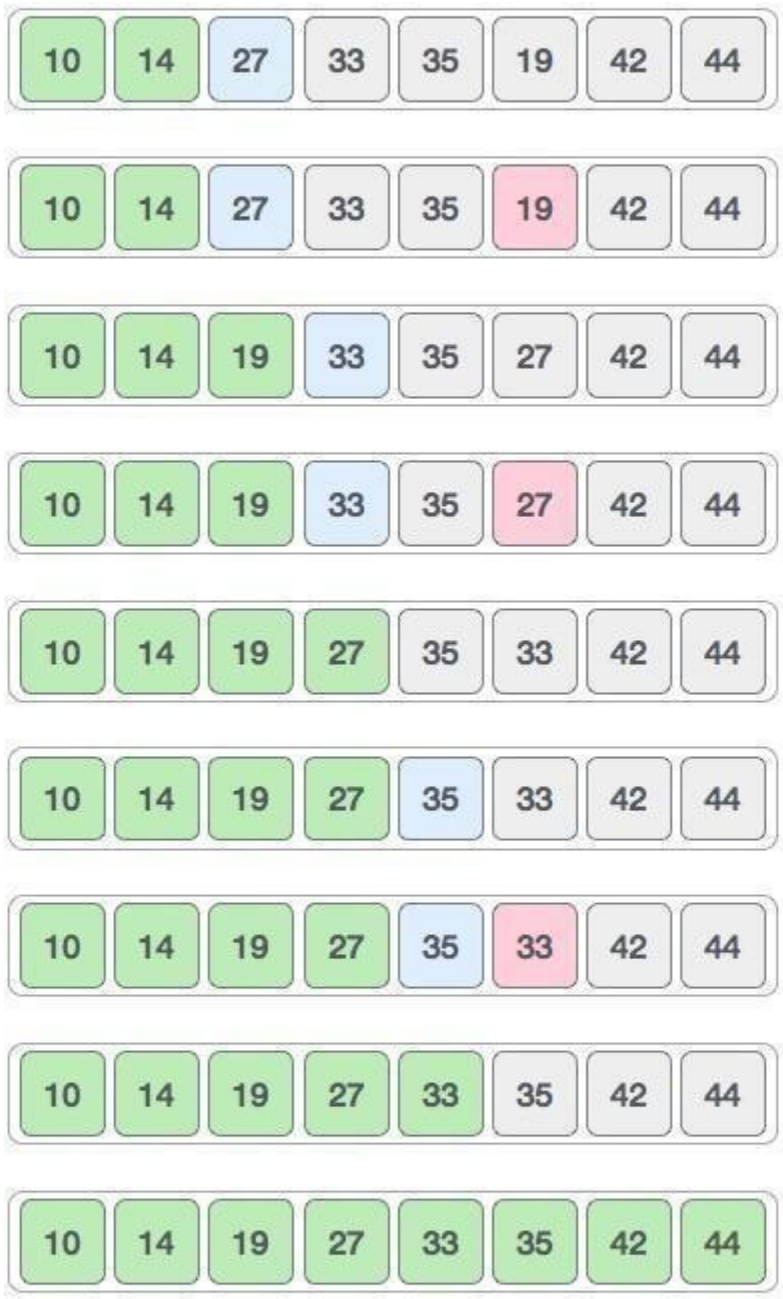
We find that 14 is the second lowest value in the list and it should appear at the second place. We swap these values.



After two iterations, two least values are positioned at the beginning in the sorted manner.



The same process is applied on the rest of the items in the array. We shall see an pictorialdepiction of entire sorting process –

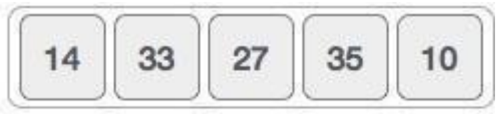


Bubble Sort:

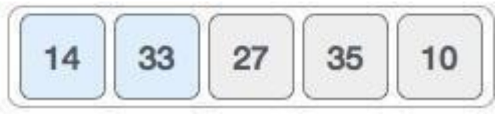
Bubble sort, sometimes incorrectly referred to as **sinking sort**, is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm gets its name from the way smaller elements "bubble" to the top of the list. Because it only uses comparisons to operate on elements, it isa comparison sort. Although the algorithm is simple, most of the other sorting algorithms are more efficient for large lists.

Working of Bubble Sort:

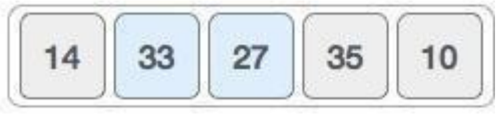
We take an unsorted array for our example. Bubble sort take O(n^2) time so we're keeping short and precise.



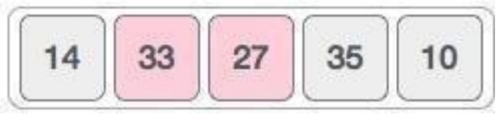
Bubble sort starts with very first two elements, comparing them to check which one is greater.



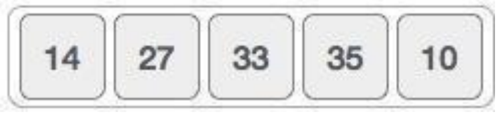
In this case, value 33 is greater than 14, so it is already in sorted locations. Next, we compare 33 with27.



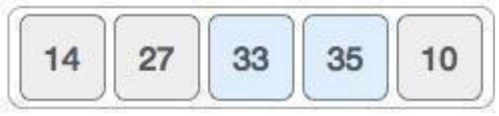
We find that 27 is smaller than 33 and these two values must be swapped.



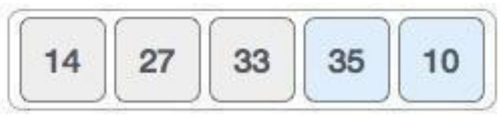
The new array should look like this −



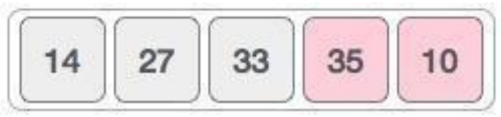
Next we compare 33 and 35. We find that both are in already sorted positions.



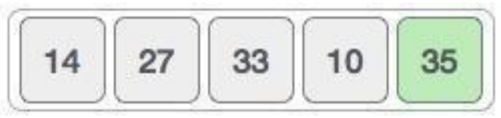
Then we move to next two values, 35 and 10.



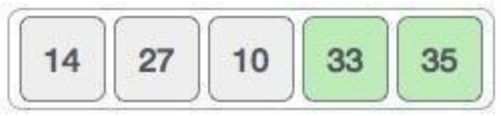
We know than 10 is smaller 35. Hence they are not sorted.



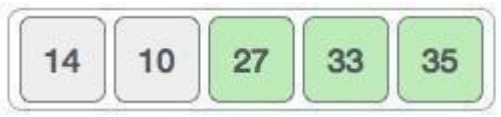
We swap these values. We find that we reach at the end of the array. After one iteration the arrayshould look like this −



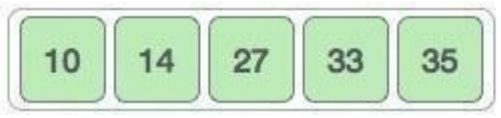
To be precise, we are now showing that how array should look like after each iteration. After second iteration, it should look like this −



Notice that after each iteration, at least one value moves at the end.



And when there's no swap required, bubble sorts learns that array is completely sorted.



Conclusion:

Hence, we learned to use selection sort and bubble sort.

Assignment No. 6

Problem Statement:

Write a python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.

Introduction:

Quick Sort:

The quick sort is an in-place, divide-and-conquer, massively recursive sort. As a normal person would say, it's essentially a faster in-place version of the merge sort. The quick sort algorithm is simple in theory, but very difficult to put into code.

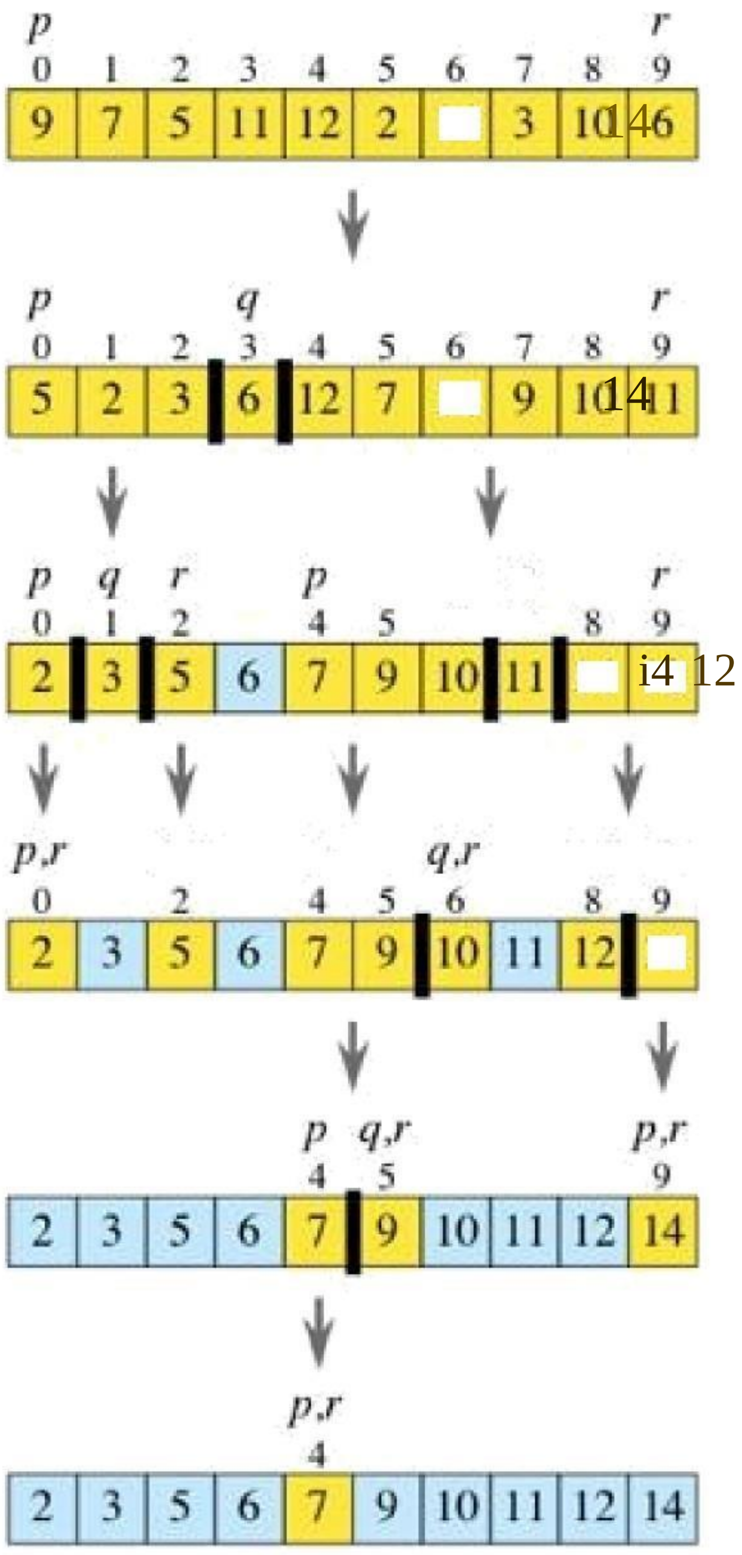
The recursive algorithm consists of four steps:

1. If there are one or less elements in the array to be sorted, return immediately.
2. Pick an element in the array to serve as a "pivot" point. (Usually the left-most elementin the array is used.)
3. Split the array into two parts - one with elements larger than the pivot and the otherwith elements smaller than the pivot.
4. Recursively repeat the algorithm for both halves of the original array.

The quick sort is by far the fastest of the common sorting algorithms.Algorithm:

1. Get N elements which are to be sorted, and store it in the array A.
2. Select the element from A[0] to A[N-1] for middle. This element is the pivot.
3. Partition the remaining elements into the segments left and right so that no elements in left has key larger than that of the pivot and no elements in right has a key smaller than that of the pivot.
4. Sort left using quick sort recursively.
5. Sort right using quick sort recursively.
6. Display the sorted array A.

Example:


p
r
0 1 2 3 4 5 6 7 8 9
9 7 5 11 12 2 3 10 14 6
q
5 2 3 6 12 7 9 10 14 11
2 3 5 6 7 9 10 11 i4 12
p,r
q,r
2 3 5 6 7 9 10 11 12
p q,r p,r
2 3 5 6 7 9 10 11 12 14

Assignment No. 7

Problem Statement:

Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of Club and last node is reserved for secretary of club. Write C++ program to maintain club member's information using singly linked list. Store student PRN and Name. Write functions to:
a) Add and delete the members as well as president or even secretary.
b) Compute total number of members of club
c) Display members
d) Two linked lists exist for two divisions. Concatenate two lists.

Introduction:

Linked List

Like arrays, Linked List is a linear data structure. Unlike arrays, linked list elements are not stored atcontiguous location; the elements are linked using pointers.

Arrays can be used to store linear data of similar types, but arrays have following limitations.

1) The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of the usage.

2) Inserting a new element in an array of elements is expensive; because room has to be created for the new elements and to create room existing elements have to shift.
For example, in a system if we maintain a sorted list of IDs in an array id[].id[] = [1000, 1010, 1050, 2000, 2040].

And if we want to insert a new ID 1005, then to maintain the sorted order, we have to move all theelements after 1000 (excluding 1000).

Deletion is also expensive with arrays until unless some special techniques are used. For example, todelete 1010 in id[], everything after 1010 has to be moved.

Advantages over arrays

1) Dynamic size

2) Ease of insertion/deletion

Drawbacks:

1) Random access is not allowed. We have to access elements sequentially starting from the firstnode. So we cannot do binary search with linked lists.

2) Extra memory space for a pointer is required with each element of the list.

Representation in C:

A linked list is represented by a pointer to the first node of the linked list. The first node is calledhead. If the linked list is empty, then value of head is NULL.

Each node in a list consists of at least two parts:

1) data

2) pointer to the next node

In C, we can represent a node using structures. Below is an example of a linked list node with aninteger data.

In Java, Linked List can be represented as a class and a Node as a separate class. The Linked Listclass contains a reference of Node class type.

Linked List vs. Array

Both Arrays and Linked List can be used to store linear data of similar types, but they both have someadvantages and disadvantages over each other.
Following are the points in favors of Linked Lists.

(1) The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance.Also, generally, the allocated memory is equal to the upper limit irrespective of the usage, and in practicaluses, upper limit is rarely reached.

(2) Inserting a new element in an array of elements is expensive, because room has to be created for thenew elements and to create room existing elements have to shifted.

For example, suppose we maintain a sorted list of IDs in an array id[].id[] = [1000, 1010, 1050, 2000, 2040, …..].

And if we want to insert a new ID 1005, then to maintain the sorted order, we have to move all theelements after 1000 (excluding 1000).

Deletion is also expensive with arrays until unless some special techniques are used. Forexample, to delete 1010 in id[], everything after 1010 has to be moved.

So Linked list provides following two advantages over arrays

1) Dynamic size

2) Ease of insertion/deletion

Linked lists have following drawbacks:

1) Random access is not allowed. We have to access elements sequentially starting from the firstnode. So we cannot do binary search with linked lists.

2) Extra memory space for a pointer is required with each element of the list.

3) Arrays have better cache locality that can make a pretty big difference in performance.

Conclusion:

Hence, we learnt different operations on linked list.

Assignment No. 8

Problem Statement:

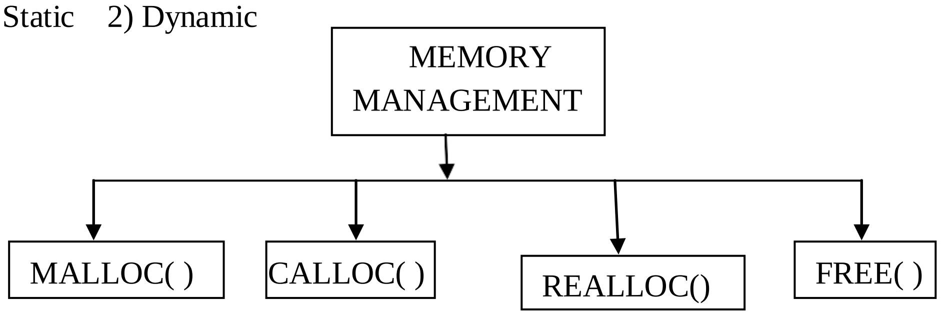
Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set B of students like butterscotch ice-cream. Write C++ program to store two sets using linked list. Compute and display
a) Set of students who like both vanilla and butterscotch
b) Set of students who like either vanilla or butterscotch or not both
c) Number of students who like neither vanilla nor butterscotch

Introduction

Dynamic Memory Management:-

In dynamic data structure, the memory space required by variables is calculated & allocatedduring execution. C gives us two choices for reserving memory locations for a variable-

1) Static 2) Dynamic



1] Block Memory Allocation (malloc()):

The malloc() function allocates a block of memory that contains the no. of bytes specified in its parameter. This returns a pointer to an area of memory with size, byte-size.

2] Contiguous Memory Allocation (calloc()):

This function can be used to allocate a contiguous block of memory, primarily for arrays. It differs from malloc() only to the extent that it sets memory to null characters. This function returns a pointer to the allocated memory.

3] Reallocation Of Memory (realloc()):

This function changes the size of the previously allocated block of memory. This is done by either deleting or extending the memory at the end of the block. If memory cannot be extended,realloc() allocates a completely new block of memory.

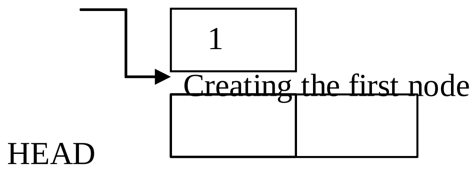
4] Releasing Memory (free()):

When the allocated memory is no longer needed, it should be returned back by using the functionfree().

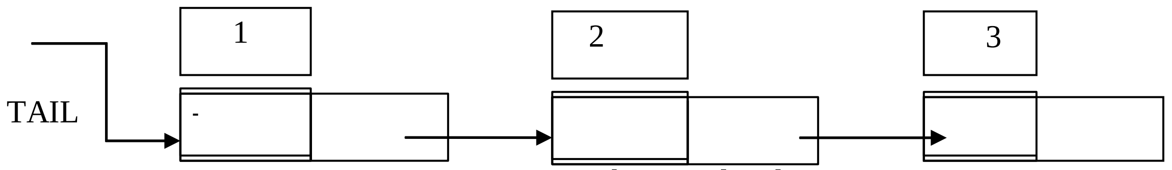
OPERATIONS ON LINK LIST:

1. CREATION:-

This operation is used to create constituent node as & when required. The structure of a list changes when nodes are inserted & deleted. Normally, creation of a list does not require alternation of the list structure except the addition of new node at the end or the beginning ofthe list.



Creating the first node



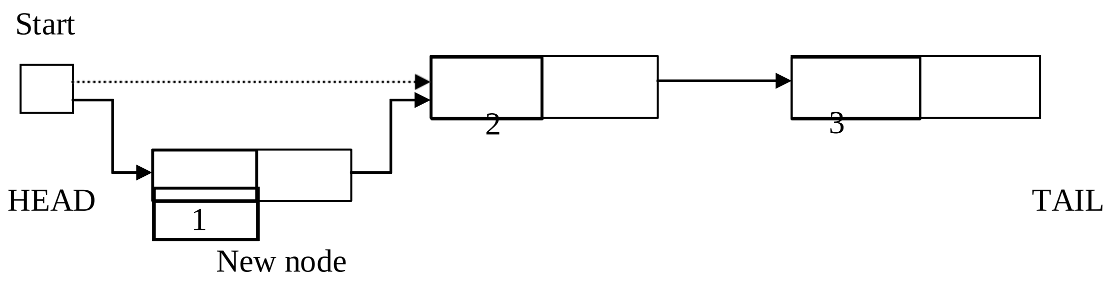
The complete list

2. INSERTION :

This operation is used to insert a new node in the link list. This can be done by three waysas:-
(i) At the beginning of the list(ii)At a certain position (iii)At the end

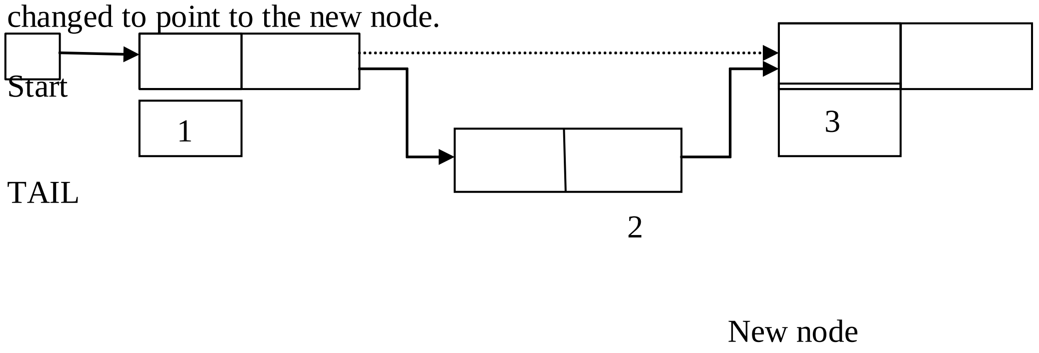
(i.)Insertion at the beginning of list:-

The insertion of a new node at the head of list is easier. The new node becomes the firstnode in the last i.e the address of the pointer head is changed to the address of new node.



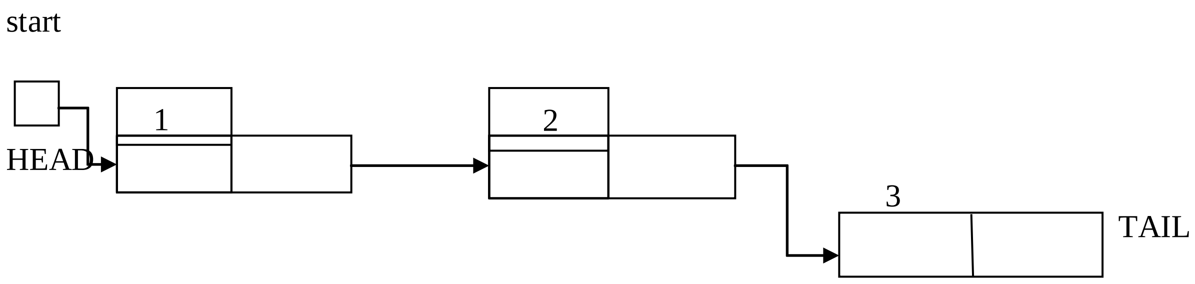
(ii.) Insertion at a certain position:-

The insertion of a data element at any point in data structure should be done in constant time. It requires two pointers to be changed. The node with data element 2 is to be inserted between data element 1 & 3. The link of new node is assigned the address of the node with data 3 & the link ofthe first node is changed to point to the new node.



(iii.)Insertion at the end of list:-

In the link list , only the head of the list is known to us. Therefore, to insert a node at the tail ofthe list, entire list has to be traversed. Once the last node is reached, the new node can be attached to the list.



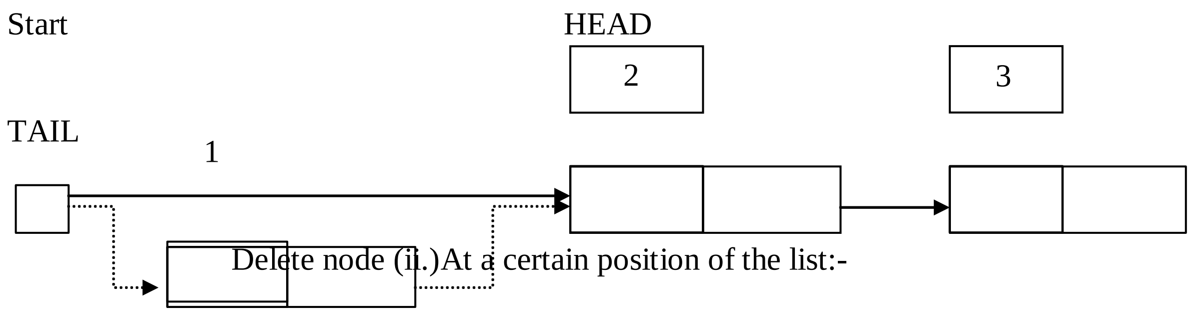
3. DELETION:-

This operation is used to delete node from the list. This can be done by three ways:-

(i) At the beginning of the list(ii)At a certain position (iii)At the end of list

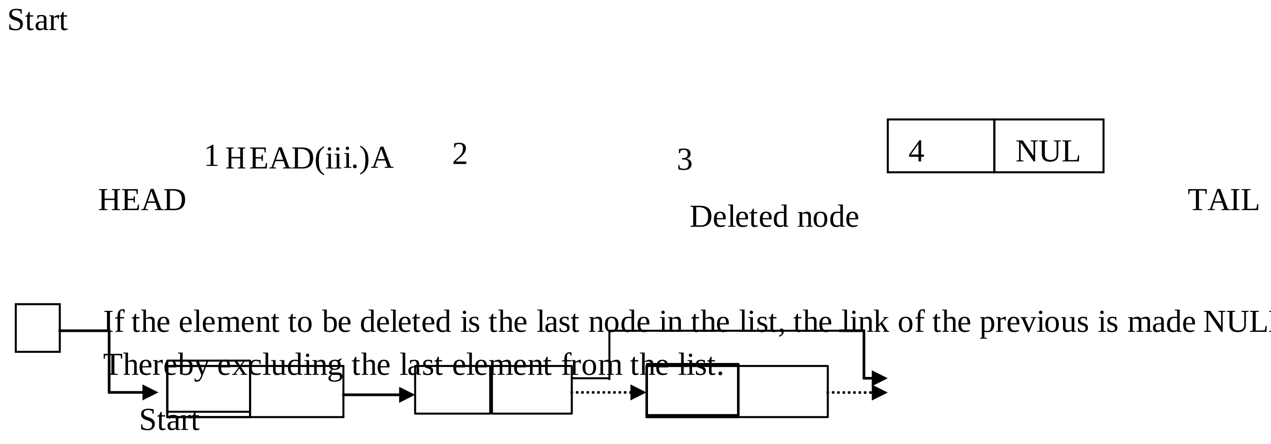
(i.)Deletion at the beginning of the list:-

If the element to be deleted is located at the first position, the deletion procedure only consists of advancing the HEAD pointer to the second element in the list.

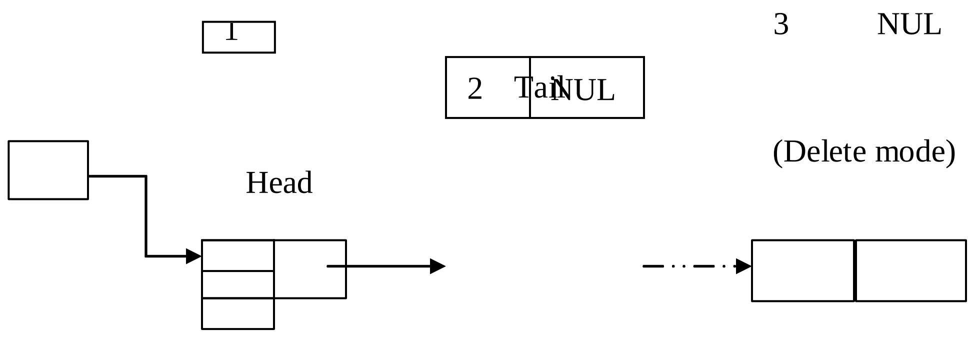


(ii.)At a certain position of the list:-

If an element in the middle of the list has to be deleted, only one pointer value has to bechanged. The value in the link field of the preceding node is changed to the address of succeeding node.



If the element to be deleted is the last node in the list, the link of the previous is made NULL. Thereby excluding the last element from the list.



4. SEARCHING:-

Searching refers to the finding of an item or data from a given list of elements .So in case of link list searching means finding a specific code from a specific list. In this operation we compare the node value to be searched with each node value in the list.

Conclusion:

Hence we have studied all the operations on link list successfully.

Assignment No. 9

Problem Statement:

A palindrome is a string of character that's the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions-

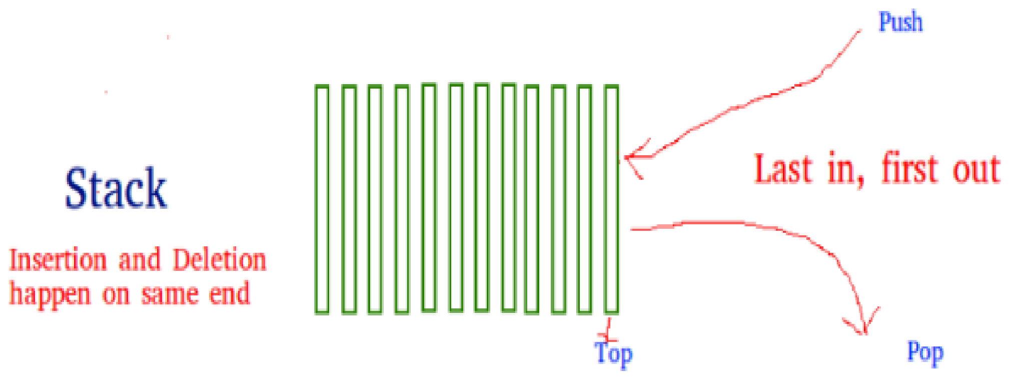
1. To print original string followed by reversed string using stack
2. To check whether given string is palindrome or not.

Introduction:

Stack Data Structure

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO (First In Last Out).
Mainly the following three basic operations are performed in the stack:

- Push: Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.
- Pop: Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
- Peek or Top: Returns top element of stack.
- isEmpty: Returns true if stack is empty, else false.



Implementation:

There are two ways to implement a stack:

- Using array
- Using linked list

Palindrome:

A palindrome is a string of character that's the same forward and backward.
Typically, punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters ―poor dan is in a droop and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical.

ALGORITHM:-

1. Start
2. Take a string as input and store it in the array s[].
3. Load each character of the array s[] into the array stack[].
4. Use variables front and top to represent the last and top element of the array stack[].
5. Using for loop compare the top and last element of the array stack[]. If they are equal, then delete the top element, increment the variable front by 1 and compare again.
6. If they are not equal, then print the output as <It is not a palindrome=.
7. Compare the elements in the steps 4-5 upto the middle element of the array stack[].
8. If every characters of the array is equal, then it is a palindrome.
9. Stop

Conclusion:

Thus we have studied different operation on stack.

ASSIGNMENT NO. 10

Problem Statement:

Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions:
1. Operands and operator, both must be single character.
2. Input Postfix expression must be in a desired format.
3. Only '+', '-', '*' and '/ ' operators are expected.

Introduction:

When you write an arithmetic expression such as B * C, the form of the expression provides you with information so that you can interpret it correctly. In this case we know that the variable B is being multiplied by the variable C since the multiplication operator * appears between them in the expression. This type of notation is referred to as **infix** since the operator is *in between* the two operands that it is working on.

Consider another infix example, A + B * C. The operators + and * still appear between the operands, but there is a problem. Which operands do they work on? Does the + work on A and Bor does the * take B and C? The expression seems ambiguous.

In fact, you have been reading and writing these types of expressions for a long time and they do not cause you any problem. The reason for this is that you know something about the operators + and *. Each operator has a **precedence** level. Operators of higher precedence are used before operators of lower precedence. The only thing that can change that order is the presence of parentheses. The precedence order for arithmetic operators places multiplication and division above addition and subtraction. If two operators of equal precedence appear, then a left-to-right ordering or associativity is used.

Let’s interpret the troublesome expression A + B * C using operator precedence. B and C are multiplied first, and A is then added to that result. (A + B) * C would force the addition of A andB to be done first before the multiplication. In expression A + B + C, by precedence (via associativity), the leftmost + would be done first.

Although all this may be obvious to you, remember that computers need to know exactly what operators to perform and in what order. One way to write an expression that guarantees there willbe no confusion with respect to the order of operations is to create what is called a **fully parenthesized** expression. This type of expression uses one pair of parentheses for each operator. The parentheses dictate the order of operations; there is no ambiguity. There is also no need to remember any precedence rules.

The expression A + B * C + D can be rewritten as ((A + (B * C)) + D) to show that the multiplication happens first, followed by the leftmost addition. A + B + C + D can be written as(((A + B) + C) + D) since the addition operations associate from left to right.

There are two other very important expression formats that may not seem obvious to you at first. Consider the infix expression A + B. What would happen if we moved the operator before the two operands? The resulting expression would be + A B. Likewise, we could move the operator to the end. We would get A B +. These look a bit strange.

These changes to the position of the operator with respect to the operands create two new expression formats, **prefix** and **postfix**. Prefix expression notation requires that all operators precede the two operands that they work on. Postfix, on the other hand, requires that its operators come after the corresponding operands. A few more examples should help to make this a bit clearer.

A + B * C would be written as + A * B C in prefix. The multiplication operator comes immediately before the operands B and C, denoting that * has precedence over +. The addition operator then appears before the A and the result of the multiplication.

In postfix, the expression would be A B C * +. Again, the order of operations is preserved sincethe * appears immediately after the B and the C, denoting that * has precedence, with + coming after. Although the operators moved and now appear either before or after their respective operands, the order of the operands stayed exactly the same relative to one another.

Table 2: Examples of Infix, Prefix, and Postfix

Infix Expression	**Prefix Expression**	**Postfix Expression**
A + B	+ A B	A B +
A + B * C	+ A * B C	A B C * +

Now consider the infix expression (A + B) * C. Recall that in this case, infix requires the parentheses to force the performance of the addition before the multiplication. However, when A + B was written in prefix, the addition operator was simply moved before the operands, + A B. The result of this operation becomes the first operand for the multiplication. The multiplication operator is moved in front of the entire expression, giving us * + A B C. Likewise, in postfix A B + forces the addition to happen first. The multiplication can be done tothat result and the remaining operand C. The proper postfix expression is then A B + C *.

Consider these three expressions again (see *Table 3*). Something very important has happened. Where did the parentheses go? Why don’t we need them in prefix and postfix? The answer is that the operators are no longer ambiguous with respect to the operands that they work on. Onlyinfix notation requires the additional symbols. The order of operations within prefix and postfixexpressions is completely determined by the position of the operator and nothing else. In many ways, this makes infix the least desirable notation to use.

Table 3: An Expression with Parentheses

Infix Expression	Prefix Expression	Postfix Expression
(A + B) * C	* + A B C	A B + C *

Table 4 shows some additional examples of infix expressions and the equivalent prefix and postfix expressions. Be sure that you understand how they are equivalent in terms of the order of the operations beingperformed.

Table 4: Additional Examples of Infix, Prefix, and Postfix

Infix Expression	Prefix Expression	Postfix Expression
A + B * C + D	+ + A * B C D	A B C * + D +
(A + B) * (C + D)	* + A B + C D	A B + C D + *
A * B + C * D	+ * A B * C D	A B * C D * +
A + B + C + D	+ + + A B C D	A B + C + D +

Infix Notation

We write expression in **infix** notation, e.g. **a-b+c**, where operators are used **in**-between operands.It is easy for us humans to read, write and speak in infix notation but the same does not go well with computing devices. An algorithm to process infix notation could be difficult and costly in terms of time and space consumption.

Prefix Notation

In this notation, operator is **prefix**ed to operands, i.e. operator is written ahead of operands. Forexample **+ab**. This is equivalent to its infix notation **a+b**. Prefix notation is also known as **PolishNotation**.

Postfix Notation

This notation style is known as **Reversed Polish Notation**. In this notation style, operator is **postfix**ed to the operands i.e., operator is written after the operands. For example **ab+**. This isequivalent to its infix notation **a+b**.

The below table briefly tries to show difference in all three notations –

S.n.	Infix Notation	Prefix Notation	Postfix Notation
1	a + b	+ a b	a b +
2	(a + b) * c	* + a b c	a b + c *
3	a * (b + c)	* a + b c	a b c + *
4	a / b + c / d	+ / a b / c d	a b / c d / +
5	(a + b) * (c + d)	* + a b + c d	a b + c d + *
6	((a + b) * c) - d	- * + a b c d	a b + c * d -

Parsing Expressions

As we have discussed, it is not very efficient way to design an algorithm or program to parseinfix notations. Instead, these infix notations are first converted into either postfix or prefix notations and then computed.

To parse any arithmetic expression, we need to take care of operator precedence andassociativity also.

Precedence

When an operand is in between two different operators, which operator will take the operand first,is decided by the precedence of an operator over others. For example −

As multiplication operation has precedence over addition, **b * c** will be evaluated firs. A table ofoperator precedence is provided later.

Associativity

Associativity describes the rule where operators with same precedence appear in an expression. For example, in expression **a+b−c**, both + and − has same precedence, then which part of expression will be evaluated first, is determined by associativity of those operators. Here, both +and − are left associative, so the expression will be evaluated as **(a+b−c)**.

Precedence and associativity, determines the order of evaluation of an expression. An operator precedence and associativity table is given below (highest to lowest) −

S.n.	Operator	Precedence	Associativity
1	Esponentiation ^	Highest	Right Associative
2	Multiplication (*) & Division (/)	Second Highest	Left Associative
3	Addition (+) & Subtraction (−)	Lowest	Left Associative

The above table shows the default behavior of operators. At any point of time in expressionevaluation, the order can be altered by using parenthesis. For example −

In **a+b*c**, the expression part **b*c** will be evaluated first, as multiplication as precedence overaddition. We here use parenthesis to make **a+b** be evaluated first, like **(a+b)*c**.

Postfix Evaluation Algorithm

We shall now look at the algorithm on how to evaluate postfix notation −

Step 1 − scan the expression from left to rightStep 2 − if it is an operand push it to stack
Step 3 − if it is an operator pull operand from stack and perform operationStep 4 − store the output of step 3, back to stack
Step 5 − scan the expression until all operands are consumedStep 6 − pop the stack and perform operation

Conclusion:

We have implement C++ program for expression conversion.

Assignment No. 11

Problem Statement:

Queues are frequently used in computer programming, and a typical example is the Creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue.

Introduction:

Priority Queue

Priority queue is an abstract data type which is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. If two elements have the same priority, they are served according to their order in the queue.

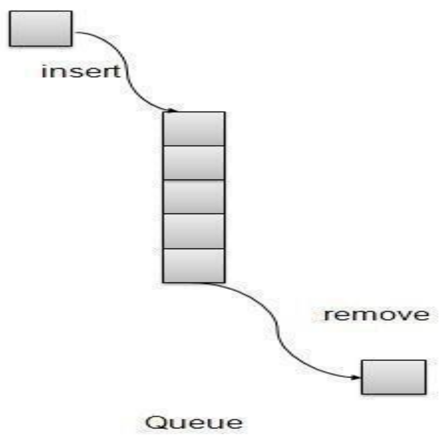
Priority Queue is more specialized data structure than Queue. Like ordinary queue, priority queue has same method but with a major difference. In Priority queue items are ordered by key value sothat item with the lowest value of key is at front and item with the highest value of key is at rearor vice versa. So we're assigned priority to item based on its key value. Lower the value, higher the priority. Following are the principal methods of a Priority Queue.

An important application of a priority queue is sorting
• Priority Queue Sort (collection S of n elements)
• put the elements in S in an initially empty priority queue by means of a series of n insert()operations on the pqueue, one for each element
• extract the elements from the pqueue by means of a series of n removeMin() operations

Basic Operations

- insert / enqueue − add an item to the rear of the queue.
- remove / dequeue − remove an item from the front of the queue.

Priority QueueRepresentation



There are few more operations supported by queue which are following.

- getHighestPriority(): Returns the highest priority item.
- Peek − get the element at front of the queue.
- isFull − check if queue is full.
- isEmpty − check if queue is empty.
-

Using Array: We're going to implement Queue using array. A simple implementation is to usearray of following structure.

```
Struct item
{
  Int item;
  Int priority;
}
```

insert() operation can be implemented by adding an item at end of array in O(1) time.

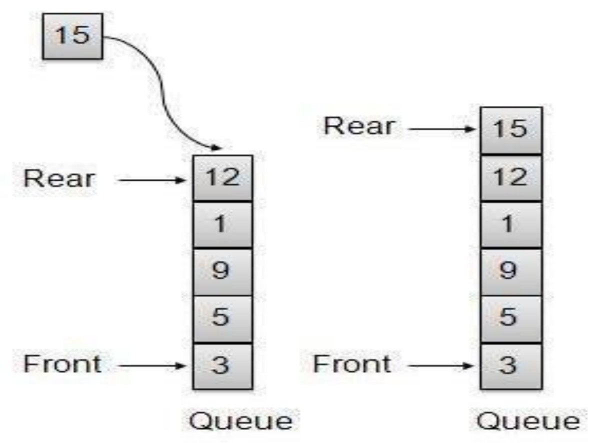
getHighestPriority() operation can be implemented by linearly searching the highest priority itemin array. This operation takes O(n) time.

deleteHighestPriority() operation can be implemented by first linearly searching an item, then removing the item by moving all subsequent items one position back.

We can also use Linked List, time complexity of all operations with linked list remains same asarray. The advantage with linked list is deleteHighestPriority() can be more efficient as we don‟thave to move items.

Insert/ EnqueueOperation

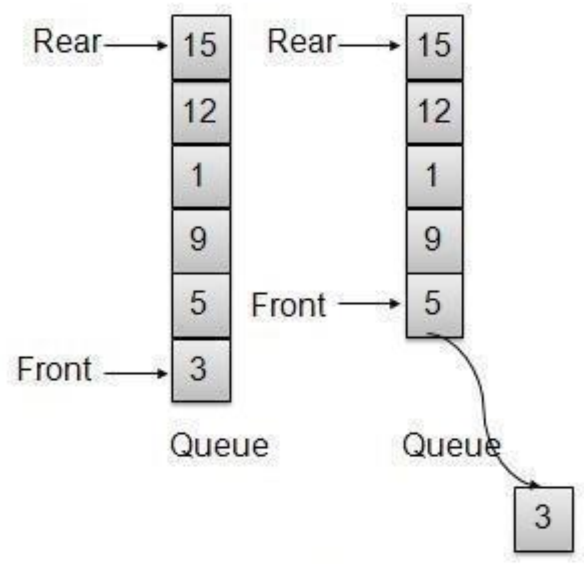
Whenever an element is inserted into queue, priority queue inserts the item according to itsorder. Here we're assuming that data with high value has low priority.



One item inserted at rear end

Remove/ DequeueOperation

Whenever an element is to be removed from queue, queue get the element using item count. Once element is removed. Item count is reduced by one.



One Item removed from front

Conclusion:

Hence, we learnt to implement a priority queue in C++ using an inorder.

Assignment No. 12

Problem Statement:

A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a onedimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque

Introductions:

Double-Ended Queue

A double-ended queue is an abstract data type similar to a simple queue, it allows you to insert and delete from both sides means items can be added or deleted from the front or rear end.

Algorithm for Insertion at rear end

```
Step -1: [Check for overflow]
            if(rear==MAX)
            Print("Queue is Overflow");
            return;
Step-2: [Insert element]
         else
            rear=rear+1;
            q[rear]=no;
            [Set rear and front pointer]

     if rear=0
            rear=1;
     if front=0
            front=1;
Step-3: return
```

Implementation of Insertion at rear end

```
void add_item_rear()
{
    int num;
    printf("\n Enter Item to insert : ");scanf("%d",&num);
    if(rear==MAX)
    {
            printf("\n Queue is Overflow");
            return;
    }
```

```
        else
        {
              rear++;
              q[rear]=num;
              if(rear==0)
                      rear=1;
             if(front==0)
                       front=1;
          }
}
```

Algorithm for Insertion at font end

```
    Step-1: [Check for the front position]
                 if(front<=1)
                      Print (“Cannot add item at front end”);
                      return;
    Step-2: [Insert at front]
                 else
                      front=front-1;
                      q[front]=no;
   Step-3 : Return
```

Implementation of Insertion at font end

```
    void add_item_front()
   {
          int num;
          printf("\n Enter item to insert:");
          scanf("%d",&num);
          if(front<=1)
          {
                 printf("\n Cannot add item at front end");
                 return;
           }
          else
          {
                      front--;
                      q[front]=num;
          }
   }
```

Algorithm for Deletion from front end

```
  Step-1 [ Check for front pointer]
             if front=0
                    print(" Queue is Underflow");
                    return;
Step-2 [Perform deletion]
          else
                    no=q[front];
                    print("Deleted element is",no);
                    [Set front and rear pointer]
                    if front=rear
                        front=0;
                        rear=0;
                    else
                        front=front+1;
Step-3 : Return
```

Implementation of Deletion from front end

```
    void delete_item_front()
    {
          int num;
          if(front==0)
          {
                    printf("\n Queue is Underflow\n");return;
          }
          else
          {
              num=q[front];
              printf("\n Deleted item is %d\n",num);
              if(front==rear)
              {
                    front=0;rear=0;
              }
              else
              {
                    front++;

              }

        }
}
```

Algorithm for Deletion from rear end

```
Step-1 : [Check for the rear pointer]
        if rear=0
                print("Cannot delete value at rear end");
                return;
Step-2: [ perform deletion]
        else
                no=q[rear];
                [Check for the front and rear pointer]
                if front= rear
                    front=0;
                    rear=0;
                else
                    rear=rear-1;
                    print("Deleted element is",no);
Step-3 : Return
```

Implementation of Deletion from rear end

```
    void delete_item_rear()
    {
        int num;
        if(rear==0)
        {
                printf("\n Cannot delete item at rear end\n");
                return;
        }
        else
        {
                num=q[rear];
                if(front==rear)
                {
                        front=0;
                        rear=0;
                }
                else
                {
                        rear--;
                        printf("\n Deleted item is %d\n",num);
                }
        }
    }
```

Conclusion:

By this way, we can perform operations on double ended queue.

Assignment No. 13

Problem Statement:

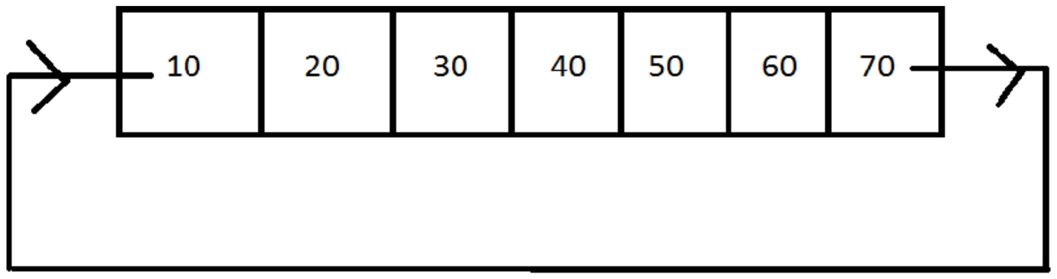
Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array.

Introduction:

CIRCULAR QUEUE:

A circular queue is an abstract data type that contains a collection of data which allows addition ofdata at the end of the queue and removal of data at the beginning of the queue. Circular queues have a fixed size. In a standard queue data structure re-buffering problem occurs for each dequeueoperation. To solve this problem by joining the front and rear ends of a queue to make the queue as a circular queue Circular queue is a linear data structure. It follows FIFO principle.

- In circular queue the last node is connected back to the first node to make acircle.
- Circular linked list fallow the First In First Out principle
- Elements are added at the rear end and the elements are deleted at front end of the queue
- Both the front and the rear pointers points to the beginning of the array.
- It is also called as “Ring buffer”.
- Items can inserted and deleted from a queue in O(1) time.



Circular Queue

Circular Queue can be created in three ways they are

- Using single linked list
- Using double linked list
- Using arrays

Using array

In arrays the range of a subscript is 0 to n-1 where n is the maximum size. To make the array as a circular array by making the subscript 0 as the next address of the subscript n-1 by using the formula subscript = (subscript +1) % maximum size. In circular queue the front and rear pointer are updated by using the above formula.

To implement a circular queue data structure using array, we first perform the following steps before we implement the actual operation:

Step 1: include all the header files which are used in program and define the constant SIZE withspecific value.

Step 2: Declare all user defined functions used in circular queue implementation.

Step 3: Create a one dimension array with above defined SIZE.

Step 4: Define two integer variables „front‟ and „rear‟ and initialize both with „-1‟ (int front=-1,rear=-1).

Step 5: Implement main method by displaying menu of operation list and make suitable fuctioncalls to perform operation selected by the user on the circular queue.

Algorithm for Enqueue operation using array

```
Step 1. start
Step 2. if (front == (rear+1)%max)
                    Print error “circular queue overflow “
Step 3. else
          { rear = (rear+1)%maxQ[rear] = element;
             If (front == -1 ) f = 0;
          }
Step 4. stop
```

Algorithm for Dequeue operation using array

```
    Step 1. start
    Step 2. if ((front == rear) && (rear == -1))
                Print error “circular queue underflow “
    Step 3. else
            { element = Q[front]
              If (front == rear) front=rear = -1


              Else
                  Front = (front + 1) % max
            }
    Step 4. Stop
```

Algorithm for create ( ) function:-

```
Step 1) allocate the memory for newnode
            newnode = new (node )
Step 2) newnode->next=newnode. // circular
Step 3) Repeat the steps from 4 to 5 until choice = „n‟
Step 4) if (root=NULL)
        root = prev=newnode // prev is a running pointer which points last node of a listelse
            newnode->next = root
            prev->next = newnodeprev = newnode
Step 5) Read the choice
Step 6)  return
```

Algorithm for display ( ) function :-

```
Step 1)  start
Step 2) declare a variable of pointer to structure node called temp, assign root to temp
            temp = root
Step 3) display temp->info
Step 4) temp = temp->next
Step 5) repeat the steps 6 until temp = root
Step 6) display temp info
Step 7)  temp=temp->next
Step 8)  return
```

<u>Conclusion:</u>

Hence, we learnt to implement circular queue using array in C++.