Questions Bank

LINKED LIST

**Q1\_1** Write a method trimEnds that could be added to the LinkedList class designed in (2) above. The method accepts an integer parameter *k* and removes *k* elements from the front of the list and *k* elements from the back of the list. Suppose a LinkedIntList variable list stores the following values:

**[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]**

The call list.trimEnds(3); would change the list to store the following elements: **[40, 50, 60, 70, 80]**

If we followed this by a second call of list.trimEnds(1); , the list would store the following elements:

**[50, 60, 70]**

If the list is not large enough to remove *k* elements from each side, give an appropriate error message. If the list contains exactly 2*k* elements, it should become empty as a result of the call. If *k* is 0 or negative, the list should remain unchanged.

**Q1-2**Write a method (named as expand) that could be added to the LinkedList class. The method accepts an integer *f* as a parameter and replaces every value *i* with *f* copies of the value (*i / f*). Suppose a LinkedIntList variable list stores the following values:

[21, 8, 15, 0, -3, 32]

The call list.expand(3); would change the list to store the following elements:

[7, 7, 7, 2, 2, 2, 5, 5, 5, 0, 0, 0, -1, -1, -1, 10, 10, 10]

If an element of the original list is not evenly divisible by *f*, as with 8 and 32 above, the resulting list should truncate any fractional component (as is done naturally by integer division). If the parameter value passed is 1, the list is unchanged. If it is 0 or negative, the list should become empty.

**Ques3-1** Read any book on C++. Make a list of syntactical differences in C and C++ procedural programming. Compile all the differences in your lab notebook and show it to your instructor.

**Ques1-4**Create a structure in C++ to implement the following Data Structures:

a) Stack

b) Queue

With members an integer array of variable size **(use only pointer no predefined size)** and necessary index locator e.g. top, front, rear etc. Declare and Define functions to represent behaviour of stack (LIFO) and Queue (FIFO) with appropriate parameters wherever required. Can you put these functions inside structures definitions, test it and give result. Can you invoke those functions using structure variables, try out all the possibilities using main method.

**QuesHAM4.** Add following items to the queue class created in **Q2**

A G D E O A Y J A J G O A F

Write a function that replaces each copy of an item in the queue with another item:

ReplaceItem(char olditem, char newitem)

For above data replace 'A' with 'a'.

**QuesHam2** Redesign the basic queue operation i.e. enqueue and dequeue using two stack class objects. Also implement it vice-versa (i.e. basic stack operations of push and pop using two queue class objects).

**Ques date+string(2 questions ).** Design a class String and Date with appropriate data members so that these can be used as standard classes for further problems.

a) Date class should provide following functionality: (1) Difference of two dates (2) Gives a next date (3) Add a number of days and get the new date (4) Display dates in appropriate formats

b) String class with size dynamically initialized and provide following functionality: (1) Find Substring from a string (2) Concatenate two strings (3) Compare two strings (4) Copy one string to another (5) Find the length of string (6) Find index of first occurance of a given character in the string.

Q3-1

Design and implement a class for representing a set of integers. The required sets contain only even numbers from 2 to 100 (both 2 and 100 included). Your class must have following constructors.

a) A constructor without any parameter that creates an empty set.

b) A parameterized constructor

c) A copy constructor

And following member methods.

a) to find union of two sets.

b) to find intersection of two sets.

c) to check whether a set is subset of another set or not

d) add an element to the set

e) remove an element from the set

Q3-2

3) Create a class student having name, roll no and semester.

• Create a default constructor

• Create a parameterized constructor with three parameters and default value 1 for semester.

• In main function create an object s1 with name Abc, Roll no. 221 and semester 3.

• Now change the semester of s1 from 3 to 4.

• Create an object s2 with values user defined.

• Create a copy constructor for the class

• Create a new object s3 with same values as s1.

• Create an array of n students where n is user defined.

Write a function check() which is global function and not a member function of class to check whether the data members of s1 and s3 are same or not.

Q3-4

Can we overload a constructor? If yes then write a program to overload a constructor with following options: (a) initialization of private data members, (b) print 10 fibonacci numbers, (b) print factorial of a number and (c) exit from the execution. Input to all these numbers are provided from main() function. Also, a menu is constructed in main function to ask option from user and depending upon the option that user enter, object of class to call the overload constructor is created.

## Q4. Add two numbers represented by linked lists | Set 2

Given two numbers represented by two linked lists, write a function that returns sum list. The sum list is linked list representation of addition of two input numbers. It is not allowed to modify the lists. Also, not allowed to use explicit extra space (Hint: Use Recursion).

Q5.

## XOR Linked List – A Memory Efficient Doubly Linked List | Set 2

In the [previous post](http://www.geeksforgeeks.org/archives/12367), we discussed how a Doubly Linked can be created using only one space for address field with every node. In this post, we will discuss implementation of memory efficient doubly linked list. We will mainly discuss following two simple functions.

Q6.

## Delete N nodes after M nodes of a linked list

Given a linked list and two integers M and N. Traverse the linked list such that you retain M nodes then delete next N nodes, continue the same till end of the linked list.

Q7

## Swap Kth node from beginning with Kth node from end in a Linked List

Given a singly linked list, swap kth node from beginning with kth node from end. **Swapping of data is not allowed, only pointers should be changed.** This requirement may be logical in many situations where the linked list data part is huge (For example student details line Name, RollNo, Address, ..etc). The pointers are always fixed (4 bytes for most of the compilers).

Q8.

## Segregate even and odd nodes in a Linked List

Given a Linked List of integers, write a function to modify the linked list such that all even numbers appear before all the odd numbers in the modified linked list. Also, keep the order of even and odd numbers same.

ARRAY

Q1. Print the pattern(DIAMOND):

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \*

\* \* \*

\*

## Q2. Divide a string in N equal parts

**Difficulty Level:** Rookie

**Question:**  
Write a program to print N equal parts of a given string

Q3.

Reverse words in a given string

Example: Let the input string be “i like this program very much”. The function should change the string to “much very program this like i”

Algorithm:

1) Reverse the individual words, we get the below string.

"i ekil siht margorp yrev hcum"

2) Reverse the whole string from start to end and you get the desired output.

"much very program this like i"

Q4.

## Write a program to reverse an array

**Iterative way:**  
1) Initialize start and end indexes.   
start = 0, end = n-1  
2) In a loop, swap arr[start] with arr[end] and change start and end as follows.  
start = start +1; end = end – 1

Q5.

## Find the Missing Number

You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in list. One of the integers is missing in the list. Write an efficient code to find the missing integer.

**Example:**

I/P [1, 2, 4, ,6, 3, 7, 8]

O/P 5

Bst

Q1. Create Bst recursively