100 Questions Challenge

Semester: III **Division: D** **Year: 2018-19**

While you all fight, Don't miss a sight,
Goal is the threshold, scores behold.

Note: Repeated questions will not be evaluated. Unnumbered will not be too.

Total Challenge Points: **2500** Threshold you need to cross: **1800**

Set A – Creative Corner

Write a C Program for the following poems [Each carries 10 Points = 80 Points]

1. Data, data, all I see,
Integer data, data of sea,
Biggest of all is what I need,
Rest of them is just a weed.

2. Take a number, in one to ten,
Take another, random one,
Play with them, as you like,
Give me unique, will stop the strike.

3. I stand out, see the sky,
Never ending, is it a lie?
Chance I get is until five,
Just done ask me what and why!

4. Strings are coming one by one,
I keep counting vowels ten,
After ten, I go to my den,
I retire and get my pension.

5. Push, Pop, Peek and Print,
Top gives the entire hint.
The first will be the last out,
Stack it is, no doubt!

6. Rear to attach,
Front to detach,
First in First out is the new,
Enqueue, Dequeue, It's the Queue.

7. malloc makes a call,
Pointer makes the move,
Node houses them all,
Thus the list is born!

8. Give the key,
Get the address.
Makes the data,
faster access.

Here are some Bonus Questions from 9 to 30. Creative and logical answers will fetch 5 points. Poor answers will fetch 1 point. Not answering will fetch -2 marks. [Considered Points = 50]

9. Reversing a palindrome string using stack, Hilarious or Geeky?
10. What do you get when you peek into an empty stack?
11. To '**understand recursion'**, you need to 'understand **recursion'**. Understand?
12. If a pointer goes dangling, what did the free() do?
13. "Being Key Address", whose t-shirt quote would it be?
14. Who is best friends with front index?
15. Why was rear index unhappy?
16. Where do you scatter, when you collide?
17. A node had two NULL's. Was it too many?

18. Who left the pointers out? Who? Who? Who?
19. Who is the major supporter of malloc()?
20. Why did circular queue graduate from college early?
21. Who would stand against 'static' in elections?
22. Why isn't linear queue not talking anymore with anyone?
23. Who terminates a struct definition?
24. Why does pointer consider itself to be a coolest dude?
25. Why does not linked list watch Hindi movies?
26. If stackie and queueie were twins, what would be their differing features?
27. Stack is growing at one end and heap at another. What happens next?
28. Is data structure male or female? Why?
29. Why are strings, array?
30. If that's a chair and scope is a table, what's that?

31. Write as many examples you can write for stacks. The example you write, if also written by other team will fetch you -5 points. If unique, will fetch you 20 points. **[Considered points: 100]**

32. Write as many examples you can write for queues. The example you write, if also written by other team will fetch you -5 points. If unique, will fetch you 20 points. **[Considered points: 100]**

33. Capture the following scenarios using structures: Keep them Nested. **[100 Points]**
a. A family candle light dinner
b. A walk along the beach
c. Spiderman rescues the city
d. System hangs. You stare at the black screen reading the error status codes
e. A ping pong ball hits the walls across the hall

34. Design a logo for the following. **[100 Marks]**
Think the following way: If you were to design a metaphor that would stand as in icon, then what would be it? Like say do you see how 'My computer' icon looks like? Do you see how 'recycle bin' looks like? In so case, do the same for following:
a. Stacks **b.** Queues **c.** List **d.** Heap

35. Talking to a kid! **[20 Points]**
How do you explain to your **5 year old** nephew who desperately wants to know 'what are data structures and algorithms'? Now, don't discourage the kid by saying it's an Engineer's headache!

36. Talking to grandma! **[20 Points]**
How do you explain to your **70 year old** grandmother who desperately wants to know 'what are data structures and algorithms'?

37. With the help of suitable code snippets, **[20 Points]**
Prove That: "Queue is **NO** more exactly a First In First Out data structure"

Set B – The Usual’s. Naturally Boring.

38. Define the following: **[20 Points]**

a. Stack
b. Queue
c. Arrays
d. Structures
e. Pointers
f. Strings
g. Heap
h. malloc
i. Linked List
j. Data Structure

[Question 39 to 50 carry 20 points each]

39. Write a program for string stack operations where top is initialized to 0.
40. Write a program for linear queue operations where front is initialized to 0 and rear to -1. Assume the data which goes inside the queue is of type float.
41. Implement a circular queue with integer data where front and rear are pointing to MAXQUEUE -1.
42. Write a program for a linear queue where both front and rear are initialized to MAXQUEUE-1.
43. Implement a double ended stack with two tops, where one is pointing to 0 and another at STACKSIZE - 1.
44. Implement a double ended circular queue for string data.
45. Implement a double ended linear queue for string data.
46. Implement a singly linked list with node holding a character data type.
47. Implement a doubly linked list with node holding a float data type.
48. Implement a circular list with node holding a string data and with a last pointer.
49. Implement a doubly linked list with a last pointer pointing to last node in the list.
50. Implement a circular list with node holding integer data and with a start pointer.

Set C – Somewhat Think-able’s

51. Iteration is better than recursion. **[20 Points]**
Comment with three to four justifying statement.

52. Optimize the code given below. **[20 Points]**
Compact the code to fewer lines.

```
Void displayQueueElements(Queue *q) {
  int i = 0;
  if(q-> front > q->rear) {
    printf(“QUEUE EMPTY\n”);
    return;
  }
  else {
    printf(“QUEUE ELEMENTS ARE\n”);
    for(i=q->front; i<=q->rear; i++) {
      printf(“%d\n”, q->items[i]);
      return;
    }
  }
}
```

53. Define backtracking. In simplest terms. **[20 Points]**

54. Write a peek function on stack which can old 10 Facebook status messages. **[20 Points]**

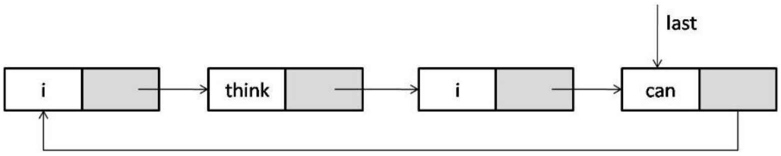
55. Recursive string length. **[20 Points]**
Write a code snippet where a string length is obtained using a recursive code.

56. Towers of Hanoi **[50 Points]**
Write a program for towers of Hanoi problem. Show the manual tracing for 2 and 3 disk problem.

57. Following are the keys and addresses generated using a hash function. Demonstrate on how the keys would be hashed if the hashing technique used was chained progressive overflow. **[50 Points]**

Key	Address Generated
Rachel	43
Monica	41
Phoebe	43
Joey	43
Chandler	41
Ross	43

58. Consider the circular list given below with string data: **[50 Marks]**



Write a function which will display the output in following fashion:
i think i can
think i can
i can
can
At each line, the function should **display** data from all the nodes present. After printing each line, an appropriate node has to be **deleted**. After printing the last line, initialize "last" to NULL

59. // **Function Name:** summon **[40 Marks]**
// **Input Params**: base address of string
// **Return Type:** base address of summoned string
// **Description:** A magician wants to generate summoning charms.
// For input string "firebolt", the function should produce "summon firebolt". Prefix the passed
// string with 'summon' and return it. For a successful magic, pointer operations are mandatory and
// usage of inbuilt string handling functions are restricted.

60. The 'N' queen problem. **[40 Marks]**
Using **state space tree** prove that:

- There is no solution for a 2 queen problem
- There are multiple solutions for a 4 queen problem

61. Explain the various collision resolution techniques. **[50 Marks]**

62. With code snippets explain iteration and recursion. You need to pick one function, write both recursive and iterative code and then explain. **[20 Marks]**

63. What is the difference between call by value and call by reference? Explain with suitable figures and codes. **[30 Marks]**

Set D – The Challenges. Mind it!

[Each question here carries 20 Points]

64. You are given a Linked List and a number K. You have to reverse it in the groups of K
Ex : [1] -> [2] -> [3] -> [4] -> [5] -> null, K = 3
output: [3] -> [2] -> [1] -> [5] -> [4] -> null
65. Write a function to get the intersection point of two Linked Lists.
66. Given two linked lists find if they are making a shape of 'Y' or a shape of 'V'.
67. Write a function to check if a singly linked list is a palindrome.
68. Construct a linked list from 2D matrix.
69. Write a code to merge two linked lists.
70. Write a function to perform union and intersection on two singly linked list.
71. Remove duplicates from a sorted linked list.
72. Write a function to reverse a list.
73. Implement a function to get the middle node of the list.
74. Consider sorted singly linked list having following nodes.
10->30->50->70->NULL
You are given pointer to node 50 and a new node having value 40. Can you inserted node 40 correctly in the list maintaining the ascending order?
75. Given number k, for Single linked list, skip k nodes and then reverse k nodes, till the end.
76. Reverse a doubly linked list without using any extra space.
77. Write a function to print alternate nodes in a linked list.
78. Implement sorting on a singly linked list.
79. Write a function to detect a loop in a singly linked list.
80. Append the last ‘n’ nodes of linked list to the beginning.
81. A list has repeating data items. Find the count of repeat of each data item in the list.
82. Find the max and min elements in the linked list.
83. Accept an integer value from the user and convert each digit into a node of list.
Example: Input: 4872
Output List: 4->8->7->2->NULL
84. Write a function to insert a node at nth position in a doubly linked list.
85. Write a function to concatenate two given lists.

Set E – MCQ’s

[Each question here carries 10 Points]

86. The postfix expression corresponding to the infix expression a + b × c - d ^ e ^ f is
A. abc × + def ^ ^ -
B. abc × + de ^ f ^ -
C. ab + c × d - e ^ f ^
D. - + a × bc ^ ^ def

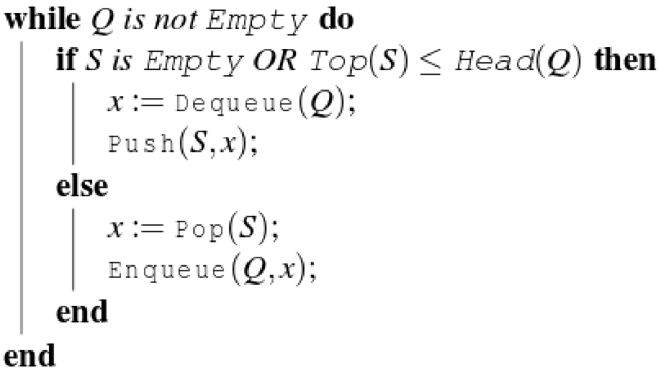
87. Following is C like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing.

```
void fun(Queue *Q) {
  Stack S;  // Say it creates an empty stack S
  // Run while Q is not empty
  while (!isEmpty(Q)) {
    // deQueue an item from Q and push the dequeued item to S
    push(&S, deQueue(Q));
  }
  // Run while Stack S is not empty
  while (!isEmpty(&S)) {
   // Pop an item from S and enqueue the poppped item to Q
   enQueue(Q, pop(&S));
  }
}
```

What does the above function do in general?

A. Removes the last from Q
B. Keeps the Q same as it was before the call
C. Makes Q empty
D. Reverses the Q

88. Let Q denote a queue containing sixteen numbers and S be an empty stack. Head(Q) returns the element at the head of the queue Q without removing it from Q. Similarly Top(S) returns the element at the top of S without removing it from S. Consider the algorithm given below.



```
while $Q$ is not $Empty$ do
    if $S$ is $Empty$ OR $Top(S) \leq Head(Q)$ then
        $x$ := Dequeue($Q$);
        Push($S$,$x$);
    else
        $x$ := Pop($S$);
        Enqueue($Q$,$x$);
    end
end
```

The maximum possible number of iterations of the while loop in the algorithm is

A. 16
B. 32
C. 256
D. 64

89. Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:

i. isEmpty (Q) — returns true if the queue is empty, false otherwise.
ii. delete (Q) — deletes the element at the front of the queue and returns its value.
iii. insert (Q, i) — inserts the integer i at the rear of the queue.

Consider the following function:

```
 void f (queue Q) {
int i ;
if (!isEmpty(Q)) {
  i = delete(Q);
  f(Q);
  insert(Q, i);
 }
}
```

What operation is performed by the above function f ?

A. Leaves the queue Q unchanged

B. Reverses the order of the elements in the queue Q

C. Deletes the element at the front of the queue Q and inserts it at the rear keeping the other elements in the same order

D. Empties the queue Q

90. Is it possible to create a doubly linked list using only one pointer with every node.

A. Not Possible

B. Yes, possible by storing XOR of addresses of previous and next nodes.

C. Yes, possible by storing XOR of current node and next node

D. Yes, possible by storing XOR of current node and previous node

91. Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing.

```
void fun(int n)
{
   Stack S;  // Say it creates an empty stack S
   while (n > 0)
   {
     // This line pushes the value of n%2 to stack S
     push(&S, n%2);
     n = n/2;
   }
   // Run while Stack S is not empty
   while (!isEmpty(&S))
     printf("%d ", pop(&S)); // pop an element from S and print it
}
```

What does the above function do in general?

A. Prints binary representation of n in reverse order

B. Prints binary representation of n

C. Prints the division result

D. Prints the division result in reverse order

92. What is the output of following function for start pointing to first node of following linked list?

1->2->3->4->5->6

```
void fun(struct node* start)
{
  if(start == NULL)
```

```
   return;
  printf("%d  ", start->data);

  if(start->next != NULL )
   fun(start->next->next);
  printf("%d  ", start->data);
}
```

A. 1 4 6 6 4 1

B. 1 3 5 1 3 5

C. 1 2 3 5

D. 1 3 5 5 3 1

93. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?

A. Delete the first element

B. Insert a new element as a first element

C. Delete the last element of the list

D. Add a new element at the end of the list

Set F – The Real Challenges

[Each question here carries 100 Points]

94. Given some resources in the form of linked list you have to cancel out all the resources whose sum up to 0(Zero) and return the remaining list.

E.g-->> 6 -6 8 4 -12 9 8 -8

the above example lists which gets cancelled :

6 -6

8 4 -12

8 -8

o/p : 9

95. Suppose there is a circle. There are N petrol pumps on that circle. Petrol pumps are numbered 0 to N-1 (both inclusive). You have two pieces of information corresponding to each of the petrol pump: (1) the amount of petrol that particular petrol pump will give, and (2) the distance from that petrol pump to the next petrol pump.

Initially, you have a tank of infinite capacity carrying no petrol. You can start the tour at any of the petrol pumps. Calculate the first point from where the truck will be able to complete the circle. Consider that the truck will stop at each of the petrol pumps. The truck will move one kilometer for each litre of the petrol.

Input Format

The first line will contain the value of N.

The next N lines will contain a pair of integers each, i.e. the amount of petrol that petrol pump will give and the distance between that petrol pump and the next petrol pump.

Output Format

An integer which will be the smallest index of the petrol pump from which we can start the tour.

Sample Input
3
1 5
10 3
3 4

Sample Output
1

Explanation
We can start the tour from the second petrol pump.

96. Given a sequence of n strings, the task is to check if any two similar words come together then they destroy each other then print the number of words left in the sequence after this pairwise destruction.
Examples:
Input : ab aa aa bcd ab
Output : 3
As aa, aa destroys each other so, ab bcd ab is the new sequence.
Input : tom jerry jerry tom
Output : 0
As first both jerry will destroy each other. Then sequence will be tom, tom they will also destroy each other. So, the final sequence doesn't contain any word.

97. Given an Integer K and a queue of integers, reverse the order of first K elements of the queue, leaving the other elements in same relative order. Ex: if Queue = {10,20,30,40,50,60,70,80,90} and K = 4 then Output = {40,30,20,10,50,60,70,80,90}

98. Given an array of non-negative integers. Find the largest multiple of 3 that can be formed from array elements. For example, if the input array is {8, 1, 9}, the output should be “9 8 1”, and if the input array is {8, 1, 7, 6, 0}, output should be “8 7 6 0”.

99. Given a number n, write a function that generates and prints all binary numbers with decimal values from 1 to n. Example n=3 Output: 1, 10, 11 n=5 Output: 1, 10, 11, 100, 101

100. There are n plants in a garden. Each of these plants has been added with some amount of pesticide. After each day, if any plant has more pesticide than the plant on its left, being weaker than the left one, it dies. You are given the initial values of the pesticide in each plant. Print the number of days after which no plant dies, i.e. the time after which there are no plants with more pesticide content than the plant to their left.

Input Format
The input contains of an integer n. The next line contains n integers, describing the array p where each p denotes the amount of pesticide in plant.

Output Format
Output an integer equal to the number of days after which no plants die.

Sample Input

```
7
6 5 8 4 7 10 9
```

Sample Output

```
2
```

Explanation
Initially all plants are alive.

Plants = {(6,1), (5,2), (8,3), (4,4), (7,5), (10,6), (9,7)}
Plants[k] = (i,j) => jth plant has pesticide amount = i.

After the 1st day, 4 plants remain as plants 3, 5, and 6 die.
Plants = {(6,1), (5,2), (4,4), (9,7)}

After the 2nd day, 3 plants survive as plant 7 dies.
Plants = {(6,1), (5,2), (4,4)}

After the 3rd day, 3 plants survive and no more plants die.
Plants = {(6,1), (5,2), (4,4)}

After the 2nd day the plants stop dying.

**** MAY THE FORCE BE WITH YOU ****