**Data Structures Assignments**

1.Bob is a magician. He has an array that consists of a number. He wants to check whether the number in the array is a magic number or not. A number is said to be a magic number if the sum of the factorial values of digits of the given number is the same as the original number. Help him to complete the below pseudocode:

**Pseudocode:**

Magic(n,a)

// a is an array

{

res:=0;

temp:= a[0];

while a[0]>0 do

{

//code line 8

fact:=1;

for num:=1 to rem do

{

fact:=fact\*num;

}

res:=res+fact;

//code line 15

}

if res = temp then

write"Magic Number";

else

write "Not Magic Number"

}

What will be the code in place of lines 8 and line 15?

**Options:**

1. rem:=a[0]//10; //line 8
2. a[0]:=a[0]%10; //line15
3. rem:=a[0]/10; //line 8
4. a[0]:=a[0]%10; //line 15
5. rem:=a[0]%10; //line 8
6. a[0]:=a[0]/10; //line 15
7. rem:=a[0]%10; //line 8
8. a[0]:=a[0]//10; //line 15

**Choices**

* 1
* 2
* 3 Correct answer
* 4

2. **Problem statement**

Ben and Jones are friends. They went to a birthday party, but they don't have any gifts to present. Ben and Jones have some special list numbers individually. They have decided to merge both of their special lists into a single list.Help them to complete this task.Complete the below pseudocode

**Pseudocode**

Function mergetwolists

Input ListNode l1, ListNode l2

{

if (l1 == null);

{

return l2;

}

else if (l2 == null);

{

return l1;

}

else if (l1.val < l2.val);

{

//code line 14

return l1;

}

else {

//code line 18

return l2;

}

}

}

What will be the code in place of lines 14 and 18?

**Options:**

1. l2.next =function merge(l1.next, l2); //line 14
2. l2.next =function merge(l1, l2.next); //line 18
3. l1.next =function merge(l1, l2.next); //line 14
4. l2.next =function merge(l1, l2.next); //line 18
5. l2.next =function merge(l1.next, l2); //line 14
6. l1.next =function merge(l1, l2.next); //line 18
7. l1.next =function merge(l1.next, l2); //line 14
8. l2.next =function merge(l1, l2.next); //line 18

**Choices**

* 1
* 2
* 3
* 4 Correct answer

**3.Problem statement**

What is the output of the following pseudocode:

**Pseudocode**

Array[]={8,-6,-4,3,-7} //for input

function(int a[])

min=INT\_MAX

min\_end=0

k = Length of Array

for l=0 to k-1

min\_end = min\_end + a[l]

if (min > min\_end)

min = min\_end

if (min\_end < 0)

min\_end = 0

return min

**Choices**

* -2
* -4 Correct answer
* -6
* None of the above

4. **Problem statement**

In data structures, what is the worst-case complexity of the following algorithm:

SORT (A)

{

n ← length [A]

For i = 1 to n do

Insert A[i] into list B[A[i]/b] ..(b<- constant)

For i = 0 to n-1 do

Sort list B with Insertion sort

Concatenate the lists B[0], B[1], . . B[n-1]..

}

**Choices**

* O(n+k)
* O(nlogn)
* O(n^2) Correct answer
* O(n^2logn)

5. **Problem statement**

Ben is a Programmer expert. He does all types of questions using different approaches. One day he tried to implement stacks using queues that perform operations like push, pop, top, and bool empty. But he missed some lines of code. Help him to complete the below pseudocode.

**Pseudocode:**

Function MyStack {

Input int q1;

Input int q2;

MyStack() {

}

Function push(int x) {

// code 11

}

Function int pop() {

int val;

while (q1 not empty()) {

//code line 19

q1.pop();

if (q1 not empty())

q2.push(val);

}

while (q2 not empty()) {

//code line 26

q2.pop();

}

return val;

}

Function int top() {

int val;

while (q1 not empty()) {

val = q1.front();

q1.pop();

q2.push(val);

}

while (q2 not empty()) {

val = q2.front();

q1.push(val);

q2.pop();

}

return val;

}

Function bool empty() {

return (q1 is empty());

}

};

What will be the code in place of line 11,19 and 26?

**Options:**

1. q1.push(); // line 11
2. val = q2.front(); //line 19
3. q1.push(q1.front()); // line 26
4. q1.push(val); // line 11
5. val = q2.front(); //line 19
6. q2.push(q2.front()); // line 26
7. q1.push(); // line 11
8. val = q2.front(); //line 19
9. q1.push(q1.front()); // line 26
10. q1.push(x); // line 11
11. val = q1.front(); //line 19
12. q1.push(q2.front()); // line 26

**Choices**

* 1
* 2
* 3
* 4 Correct answer

**6.Problem statement**

Smith is a mathematician. He always loves to work with numbers. One day he heard about the concept of the palindrome. He tried to solve this. But before going to check he wanted to find the reverse order of the list. Help him to find the reverse of the given linked list. Complete the below pseudocode to help him:

**Pseudocode:**

Function reversedlist(listnode,k):

Input ListNode,k

check length of node <k if true return node

i = 0

s = listnode

while s is not None:

{

s = s.next

i = i + 1

}

if i < k:

//code line 12

head=listnode.next

reverse=listnode

reverse.next=None

count=0

while head is not None:

{

if count==k-1:

break

temp=head

//code line 22

//code line 23

reverse=temp

count+=1

}

temp1=reverse

temp2=temp1

while temp1 is not None:

{

temp2=temp1

temp1=temp1.next

}

temp2.next=reverseList(head,k)

return reverse

What will be the code in place of lines 12, 22, and 23?

**Options:**

1. return listnode //code line 12
2. head.next=head.reverse //code line 22
3. temp.next=reverse //code line 23
4. return listnode //code line 12
5. head=temp.next //code line 22
6. temp.next=reverse //code line 23
7. return listnode //code line 12
8. head=head.next //code line 22
9. temp.next=head.reverse //code line 23
10. return listnode //code line 12
11. head=head.next //code line 22
12. temp.next=reverse //code line 23

**Choices**

* 1
* 2
* 3
* 4 Correct answer

7. **Problem statement**

John is working on arrays. He has an array of integers A. A represents a histogram i.e A[i] denotes the height of the ith histogram's bar. The width of each bar is 1. He wanted to find the area of the largest rectangle formed by the histogram using Stacks. Help him to solve by completing the below pseudocode:

**Pseudocode**

function largestRectangleArea

input A,n,Left[n],right[n],stack s;

{

if(size of A is equal to 1) the

return A[0];

n = A.size();

for(int i = 0 to n,step i by 1)do

{

if(stack is empty)do

{

left[i] = -1;

}

else{

while(stack not empty and A[s.top()] >= A[i])do

{

s.pop();

}

if(stack empty)then

{

left[i] = -1;

}

else do

{

left[i] = s.top();

}

}

s.push(i);

}

while(stack not empty)

{

s.pop();

}

for(int i = n-1;i>=0; step i by -1)do

{

if(stack is empty)

{

right[i] = n;

}

else do {

//\*\*\*\*\*\*\*\*\*\*\*\*\*code line 39\*\*\*\*\*\*\*\*\*\*\*\*\*//

{

s.pop();

}

if(stack is empty) then

{

right[i] = n;

}

else{

right[i] = s.top();

}

}

s.push(i);

}

int ans = 0;

for(int i = 0 to n; step i by 1){

//\*\*\*\*\*\*\*\*\*\*code line 55\*\*\*\*\*\*\*\*//

}

return answer;

}

What will be the code in place of lines 39 and 55?

**Options**

1. while(stack not empty and A[s.top()] <= A[i]) //line 39
2. answer = max(answer,(right[i] - left[i]-1)); //line 55
3. while(stack not empty and A[s.top()] >= A[i]) //line 39
4. answer = max(answer,(left[i] - right[i]-1)]); //line 55
5. while(stack not empty and A[s.top()] <= A[i]) //line 39
6. answer = max(answer,(left[i] - right[i]-1)\*A[i]); //line 55
7. while(stack not empty and A[s.top()] >= A[i]) //line 39
8. answer = max(answer,(right[i] - left[i]-1)\*A[i]); //line 55

**Choices**

* 1
* 2
* 3
* 4 Correct answer

8. **Problem statement**

Alice and her friends were playing a game. They were given a binary search tree with K nodes and a starting root of the tree. Alice was asked to figure out a way to find the Mth smallest element present. While writing the code some of the lines went missing.

int findele(Node \*ref, int M)

int c = 0

int Ms = INT\_MIN

Node \*curr = root

while (curr != NULL)

if (curr->left == NULL)

c++;

if (c==M)

Ms = curr->key

curr = curr->right

else

Node \*last = curr->left

while (last->right != NULL && last->right != curr)

last= last->right

if (last->right==NULL)

//LINE 6

//LINE 7

else

last->right = NULL

c++;

if (c==k)

Ms = curr->key

curr = curr->right

return Ms

Can you help Alice and her friends in finding the code for line 6 and line 7?

**Options**

1. last->right = curr;
2. curr = curr->left;

1. last->left = curr;
2. curr = curr->left;

1. last->right = curr;
2. curr = curr->right;

1. last->left = curr->right;
2. curr->right = curr->left;

**Choices**

* 1 Correct answer
* 2
* 3
* 4

9. **Problem statement**

Ben is working on a problem. Then help him to find which of these algorithms is inferred from the following code using greedy analysis.

Greedy...(P, k)

{

G = empty

for each u in P do

d[u] = INFINITY

for (i = 1 to k)

{

Let u be the point of P such that d[u] is maximum

Add u to G

for (each v in P)

{

d[v] = min(d[v], distance(v,u))

}

D = max\_{v in P} d[v]

}

return (G, D)

}

**Choices**

* Prim's algorithm
* Huffman analysis
* k-center problem Correct answer
* Interval scheduling

**Problem statement**

Consider Ben is working on Data Structures. He is given two strings X and Y. He needs to find the maximum length of the common subsequence both in X and Y.

**Example**  
Consider strings X = "BHHUBC" and Y = "HYUYBZC"  
The longest Common Subsequence is “HUBC” of length 4.

So he implemented a function Length that takes the X and Y as input as m,n where m is the length of X, n is the length of Y, and returns the length of the Longest Common Subsequence.

**Pseudocode**

FUNCTION Length( X, Y, m, n )

DECLARE L = ARRAY[m+1][n+1] of INT

INT i

INT j

FOR i -> 0 to m

FOR j -> 0 to n

IF (i == 0 || j == 0)

L[i][j] = L[i-1][j-1] + 1 // Line 8

ELSE IF (X[i-1] == Y[j-1])

L[i][j] = max(L[i-1][j], L[i][j-1]) // Line 10

ELSE

L[i][j] = 0 // Line 12

ENDIF

ENDFOR

ENDFOR

RETURN L[m][n]

ENDFUN

**Note:**

* The function max() returns the maximum element. Eg max(2,4) returns 4
* In the above pseudocode, the String starts from index 1

But the above pseudocode is not correct. But if he rearranges Lines 8, 10, and 12 then the pseudocode will be correct in this scenario. Which of the following options is the correct representation of rearranging the code snippets at Lines 8, 10, and 12?

**Choices**

* 8, 12, 10
* 10, 12, 8
* 12, 8, 10 Correct answer
* 12, 10, 8