



**FACULTY OF ENGINEERING  
SCHOOL OF COMPUTING AND IT  
B. Tech (CS/IT/CCE) III Semester  
I Sessional Examination - 2019-20  
September 2019 (July-Nov 2019 Session)  
CS1303– Data Structures- Solution  
(CLOSED BOOK)**

**Duration: 1 Hour**

**Max. Marks: --15**

**Instructions:**

---

- Attempt all questions.
  - Missing data, if any, may be assumed suitably.
  - Function/Programs shall be written in programming language 'C'
- 

1. Write a program to find the largest among n integers ( Use Dynamic Memory Allocation method for storing integers in the array) using the technique described as under: [5]
- Divide the array of integers into two halves.
  - Find the largest integer in each of the sub-arrays.
  - Compare the largest integers of the sub-arrays and display the largest integer.

Write separate function for reading the array and for finding the largest integer in a sub-array. Write single function for finding the largest integer in sub-arrays.

(Hint: Pass the array, its lower and upper bounds to the function)

Ans:

```
void getdata(int *pv, int n)
{
    int i;
    for(i=0;i<n;i++)
        scanf("%d",pv+i);
}
int largest(int *pv, int lb , int ub)
{
    int i,l;
    l=*(pv+lb);
    for(i=lb;i<ub;i++)
    {
        if (*(pv+i)>l)
            l=*(pv+i);
    }

    return l;
}
```

```
void main()
```

```
{
    int n, *pv,l1,l2;
    printf("How many numbers");
    scanf("%d",&n);
    pv=(int *)malloc(n*sizeof(int));
    getdata(pv,n);
    l1=largest(pv,0,n/2);
    l2= largest(pv,n/2,n);
    if (l1>l2)
        printf("Largest is %d",l1);
    else
        printf("Largest is %d",l2);
}
```

2. A matrix of size M x N contains binary values only (0 or 1). Write a function to display row numbers in which all the values are zero. [3]

**Ans :**

```
void printrows(int A[][10], int m, int n)
{
    int i, j, flag;
    printf("Rows Containing all values as zero are :");
    for(i=0;i<m;i++)
    {
        flag=1;
        for(j=0;j<n;j++)
        {
            if (A[i][j]==1)
            {
                flag=0;
                break;
            }
        }
        If (flag==1)
            printf("%d ",i);
    }
}
```

3. Write a function/algorithm to delete odd nodes (Node number 1,3,5...) from the singly linked list. [5]

**Ans**

```
NODE * DeleteNodes(NODE *start)
{
    NODE *temp, *start1;
    start1=start->next;
    while (start!=NULL)
    {
```



```
temp=start;
start=start->next;
free(temp);
temp=start;
if (start!=NULL)
    start=start->next;
if (temp!=NULL && start!=NULL)
    temp->next=start->next;
}
return start1;
}
```

4. Consider a two dimensional array [2]

int A[100][50]; // assume that an integer requires 4 bytes.

Starting address of the array is 3500

What is the address of the element A[3][2], if the allocation is row major ?

**Ans:**

Address of element A[3][2] will be

Base Address + (3 \* 50 + 2) \* 4 = 3500 + 608 = 4108