# Answers of Some Important Viva-voce Questions on Data Structure & Algorithm

Dr. Rahul Das Gupta

## Structure of nodes present in the Linked List:

```
typedef struct s_linked_list
{
    int data;
    struct s_linked_list *nxt;
} NODE;
```

## Structure of nodes present in the Binary Search Tree:

```
typedef struct bs_tree
{
    int data;
    struct bs_tree *left, *right;
} BS_TREE;
```

**** Minutely look at the parts highlighted with yellow colour. Most of the students overlooked these important parts.**

1. **Count the even integers present in the linked list using a recursive function.**

   **Ans:**

   ```
   int count_even(NODE *p)
   {
      if (p==NULL)
          return 0;
      else if (p->data%2==0)
          return (1+count_even(p->nxt));
      else
          return count_even(p->nxt);
   }
   ```

2. **Write a recursive C function to determine the sum of odd integers present in the linked list.**

   **Ans:**

   ```
   int sum_odd(NODE *p)
   {
      if (p==NULL)
          return 0;
   ```

```
     else if (p->data%2==1)
       return (p->data+sum_odd(p->nxt));
     else
           return sum_odd(p->nxt);
}
```

## 3. Write a recursive C function to find the maximum integer present in the linked list.

**Ans:**

```
int maximum(NODE *p)
{
     int m;
     if (p==NULL)
    {
        printf("\n Empty list…");
        exit(1);
    }
    else if (p->nxt==NULL)
          return p->data;
    else
     {
        m=maximum(p->nxt);
        return (m>p->data)? m : p->data;
      }
}
```

## 4. Write a recursive C function to determine the sum of integers present in the binary search tree.

**Ans:**

```
int sum_tree(BS_TREE *p)
{
   if (p==NULL)
       return 0;
   else
     return (sum_tree(p->left) + p->data + sum_tree(p->right));

}
```

## 5. Write a recursive C function to determine the sum of the odd integers present in the binary search tree.

**Ans:**

```
int sum_odd(BS_TREE *p)
{
   if (p==NULL)
       return 0;
   else if (p->data%2==1)
     return (sum_odd(p->left) + p->data + sum_odd(p->right));
   else
       return (sum_odd(p->left) + sum_odd(p->right));
}
```

## 6. Write a recursive C function to count the odd integers present in the binary search tree.

**Ans:**

```
int sum_odd(BS_TREE *p)
{
   if (p==NULL)
       return 0;
   else if (p->data%2==1)
     return (count_odd(p->left) + 1 + count_odd(p->right));
   else
       return (count_odd(p->left) + count_odd(p->right));
}
```