

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
TEST-1 EXAMINATION (FEB 2018)
B-Tech (2nd SEM)

Course Code: 10B11CI211
Course Name: DATA STRUCTURES
Course Credit: 4

Max. Marks: 15
Max. Time: 1 HRS

Note: All questions are compulsory. Skip syntax error if there any.

Q1.

(5x1=5)

- a. What is the difference between declaration and definition of a variable ?
- b. How is an Array different from Linked List?
- c. A quadratic algorithm with processing time $T(n) = cn^2$ spends $T(N)$ seconds for processing N data items. How much time will be spent for processing $n = 5000$ data items, assuming that $N = 100$ and $T(N) = 1\text{ms}$?
- d. What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?
- e. What will be output of following C programs?

```

/*Program2
Assume integer occupies 2 bytes
*/
#include<stdio.h>
int main()
{
    struct employee
    {
        int empid[5];
        int salary;
        employee *s;
    }emp;
    printf("%d %d", sizeof(employee),
    sizeof(emp.empid));
    return 0;
}

//Program1
#include<stdio.h>
int main()
{
    printf("%d", sizeof(void *));
    return 0;
}

```

Q2. Do the worst case analysis for following given codes. Show all the computation steps.

(2x2 = 4)

```

//Prog1
void function(int n)
{
    int i, j, k, count=0;
    for(i=n/2; i<=n; i++) {
        for(j=1; j+n/2<=n; j=j+1) {
            for(k=1; k<=2; k=k*2) {
                count++;
            }
        }
    }
}

```

```

//prog2
void function(int n)
{
    int i=1, s=1;
    while(s<=n)
    {
        i++;
        s=s+i;
        printf("*");
    }
}

```

Q3.

(2 x3 =6)

- a) Design the decomposition tree to solve Tower of Hanoi problem, consisting of 3 towers A, B, & C, and no. of disks $n=4$. Also design the recursive algorithm that apply the solution for this by consisting of base condition(s), de-composition condition(s) and re-composition condition(s). **(Note: Do not write C program code to solve this problem using recursion)**
- b) Check whether the given linked list is NULL-terminated or not. If there is a cycle find the start node of the loop.
- c) Given a linked list, find whether the linked list contains a loop or not. Write a program to Detect a loop in a linked list

JUT T1 EXAMINATION 2018