

CSE- 3004 LAB-6 Assignment**Academic year:** 2020-2021**Semester:** WIN**Faculty Name:** Dr. D Sumathi Mam**Date:** 2 /9/2021**Student name:** M.Taran**Reg. no.:** 19BCE7346

1.) John and Peter went for higher studies in Australia. John bought a car and requested peter to find a fancy number for his car. Peter had a puzzle for john to solve . The constraints were the vehicle number must have 3 digits. The difference between the first two digits should be 1. If a third digit has to be formed, it must be either 1 lesser or greater than the last digit of that number. Now what all numbers can be fancy if john has to choose between 1-5. Write down the Java program to implement this using dynamic programming.

CODE :

```
import java.util.*;
public class Main {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int car[] = new int[3];
        for (int i = 1; i < 5; i++) {
            car[0] = i;
            int count1 = 0;
            for (int j = 0; j < 2; j++) {
                if (count1 == 0) {
                    car[1] = car[0] + 1;
                    count1 = 1;
                } else {
                    car[1] = car[0] - 1;
                }
            }
            int count2 = 0;
            for (int k = 0; k < 2; k++) {
                if (count2 == 0) {
```

```
        car[2] = car[1] + 1;
        count2 = 1;
    } else {
        car[2] = car[1] - 1;
    }
    boolean condition = true;
    for (int l = 0; l < 3; l++) {
        if (!(car[l] > 0 && car[l] <= 5)) {
            condition = false;
        }
    }
    if (condition) {
        System.out.println(car[0] + " " + car[1] + " " + car[2]);
    }
}
}
}
}
```

OUTPUT :

Result

compiled and executed in 1.032 sec(s)

```
1 2 3
1 2 1
2 3 4
2 3 2
2 1 2
3 4 5
3 4 3
3 2 3
3 2 1
4 5 4
4 3 4
4 3 2
|
```

2. Write a creative story that could implement 0-1 knapsack problem. Implement it using a Java program.

CODE:

```
import java.util.Scanner;
public class Main {
    static int max(int a, int b) {
        return (a > b) ? a : b;
    }
    static int main(int W, int wt[], int val[], int n) {
        int i, w;
        int[][] K = new int[n + 1][W + 1];
        for (i = 0; i <= n; i++) {
            for (w = 0; w <= W; w++) {
                if (i == 0 || w == 0)
                    K[i][w] = 0;
                else if (wt[i - 1] <= w)
                    K[i][w] = max(val[i - 1] + K[i - 1][w - wt[i - 1]], K[i - 1][w]);
                else
                    K[i][w] = K[i - 1][w];
            }
        }
        return K[n][W];
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no.of items : ");
        int n = sc.nextInt();
        System.out.println("Enter "+n+" weights : ");
        int wt[] = new int[n];
        for (int i = 0; i < n; i++) {
            wt[i] = sc.nextInt();
        }
        System.out.println("Enter "+n+" values : ");
        int val[] = new int[n];
```

```
for (int i = 0; i < n; i++) {  
    val[i] = sc.nextInt();  
}  
System.out.println("Enter the maximum capacity weight: ");  
int W = sc.nextInt();  
System.out.println("knapsack capacity obtained is: " + main(W, wt, val,  
n));  
sc.close();  
}  
}
```

OUTPUT :

Result

compiled and executed in 20.996 sec(s)

```
Enter no.of items :  
4  
Enter 4 weights :  
2  
3  
1  
4  
Enter 4 values :  
6  
7  
2  
1  
Enter the maximum capacity weight:  
5  
knapsack capacity obtained is: 13
```