ชื่อ-นามสกุล..... รหัสนักศึกษา .....

ASSIGNMENT #1 *(5 คะแนน)*

1. จงแปลงนิพจน์ Infix ในแต่ละข้อต่อไปนี้ ให้อยู่ในรูปของนิพจน์ Prefix และ Postfix

ข้อที่	Infix Notation	Prefix Notation	Postfix Notation
1.1)	(9 + 3) / 4 – (3 * 4)		
1.2)	E * D + (C - F / A) / G		
1.3)	(D + 9 * 4) + C - E / 5		
1.4)	C – (A * C - D * E) * F		
1.5)	(B + A - E * D) * (C - F) / G		
1.6)	5 * 3 + 6 - (10 / 2) + 8		

2. จงแสดงขั้นตอนการแปลงนิพจน์ Infix ให้อยู่ในรูปของนิพจน์ Postfix โดยใช้สแตก

2.1) **C * E - (A / 5 + B) + 4**

Token	Stack →	Output

2.2) **B - C * E + 2 - 3 * E**

Token	Stack →	Output

2.3) **(2 * D - (A * C - D + E)) / 5**

Token	Stack →	Output

3. จงแสดงขั้นตอนการหาผลลัพธ์จากนิพจน์ Postfix โดยใช้สแตก

3.1) **7 5 * 4 + 4 3 * + 5 -**

Token	Stack →	อธิบาย

3.2) **8 5 8 + 3 2 * - + 7 +**

Token	Stack →	อธิบาย

3.3) **2 7 5 - 5 * + 9 3 - 5 * + 8 -**

Token	Stack →	อธิบาย

4. จงแสดงกระบวนการทำงานของฟังก์ชันแบบ Recursive และผลลัพธ์ที่ได้จากฟังก์ชันต่อไปนี้

```
4.1    int Ex1(int x, int y)
       {
          if (x == 1)
              return (y + 2);
          else
              return (y + Ex1(x-1,y));
       }
```

เมื่อเรียกใช้ฟังก์ชัน ด้วยคำสั่ง `ans = Ex1(4,5);`

แสดงการทำงานของฟังก์ชัน	ผลลัพธ์

```
4.2    void Ex2(int a, float b)
       {
          if (a == 1)
              printf("B = %.2f\n",b);
          else
            {
              b = b + 1.5;
              printf("B = %.2f\n",b);
              Ex2(a-1,b);
            }
       }
```

เมื่อเรียกใช้ฟังก์ชัน ด้วยคำสั่ง `Ex2(4,25.5);`

แสดงการทำงานของฟังก์ชัน	ผลลัพธ์