QUIZ 2
Compilation Techniques

1. *Bottom-up Parsing*
   a. Diberikan *Context Free Grammar* (CFG) berikut:

```
E -> E + T | E - T | T
T -> T * F | F
F -> id | id [ E ]
```

   Berdasarkan *grammar* di atas:
   1) Buatlah diagram transisi *operasi* **GOTO (*Canonical Set*)**!

   2) Buatlah **tabel SLR** berdasarkan diagram transisi yang didapatkan dengan follow sebagai berikut:

   Follow E = {$, +, -,]}

   Follow T = {$, +, -,], *}

   Follow F = {$, +, -,], *}

   b. Diketahui tabel SLR berdasarkan CFG berikut ini:

```
S -> u X Y
X -> Y u
Y -> h | w | y
```

	Action					**GOTO**		
	u	**h**	**w**	**y**	**$**	**S**	**X**	**Y**
0	S2	S4	S5	S6		1		3
1					accepted			
2		S4	S5	S6			7	3
3	S8							
4	R3				R3			
5	R4				R4			
6	R5				R5			
7		S4	S5	S6				9
8		R2	R2	R2				
9					R1			

   Berdasarkan tabel SLR di atas, ilustrasikan pergerakan *parsing* dengan implementasi *stack* untuk *input string*: **uwuh**

2. Diberikan *statement*: **4 ^ 2 + cos(0)**
   Buatlah ***Annotated Parse Tree*** berdasarkan *Syntax Directed Definition* berikut ini:

Production	***Semantic Rules***
S → B A	{A.i = B.val} {S.val = A.s}
A → + B A_1	{A_1.i = A.i + B.val} {A.s = A_1.s}
A → ε	{A.s = A.i}
B → D C	{C.i = D.val} {B.val = C.s}
C → ^ D C_1	{C_1.i = C.i ^ D.val} {C.s = C_1.s}
C → ε	{C.s = C.i}
D → cos(E)	{D.val = cos(E.val)}
D → (S)	{D.val = S.val}
D → num	{D.val = num.val}
E → num	{E.val = num.val}

   Catatan: nilai dari cos(0) adalah 1.

3. Diberikan pernyataan:

```
a = (a-b) - (-(c+d)) * (a-b) / (c+d) * d
```

   Buatlah *intermediate code* dalam bentuk:
   a. ***Three Address Code*** **(TAC)**
   b. ***Quadruples***
   c. ***Triples***
   d. ***Directed Acyclic Graph*** **(DAG)**
4. Buatlah ***generated machine code*** dari penggalan kode berikut ini:

```
int b = 0;
for(int a = 1; a < 25; a++){
  if (a % 3 == 1){
          b+=a;
  }
}
x = b;
```