

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
\*\*\*\*\* \*\*\*\*\*



**BÀI THỰC HÀNH SỐ 7: SINH MÃ - CODE GENERATION**  
**HỌC PHẦN: THỰC HÀNH XÂY DỰNG CHƯƠNG TRÌNH DỊCH**

Mã lớp học: 161629

GVHD: Nguyễn Thị Thu Hương

TA: Đỗ Gia Huy

Sinh viên thực hiện: Trịnh Hữu An - 20225593

- Hà Nội, tháng 1 năm 2026 -

# Mục lục

<b>1 Phân tích chi tiết Example 1 - Chương trình không có chương trình con</b>	<b>3</b>
1.1 Source Code . . . . .	3
1.2 Mô tả chương trình . . . . .	3
1.3 Generated Code . . . . .	3
1.4 Phân tích chi tiết từng phần . . . . .	4
1.4.1 Khởi tạo Stack Frame . . . . .	4
1.4.2 Khởi tạo biến S := 0 . . . . .	4
1.4.3 Khởi tạo biến I := 1 . . . . .	5
1.4.4 Vòng lặp WHILE I <= 5 . . . . .	5
1.4.5 In kết quả (Instructions 25-27) . . . . .	6
1.4.6 Kết thúc chương trình (Instruction 28) . . . . .	7
<b>2 Phân tích chi tiết Example 3 - Chương trình có chương trình con</b>	<b>8</b>
2.1 Source Code . . . . .	8
2.2 Mô tả chương trình . . . . .	8
2.3 Generated Code . . . . .	9
2.4 Phân tích chi tiết . . . . .	11
2.4.1 Phần PROCEDURE HANOI (Instructions 0-40) . . . . .	11
2.4.2 Phần Main Program (Instructions 41-102) . . . . .	12

# 1 Phân tích chi tiết Example 1 - Chương trình không có chương trình con

## 1.1 Source Code

Listing 1: example1.kpl - Chương trình tính tổng bình phương

```
1 PROGRAM CODEGEN9;
2 TYPE T = INTEGER;
3 VAR S : T;
4     I : INTEGER;
5
6 BEGIN
7     S := 0;
8     I := 1;
9     WHILE I <= 5 DO
10    BEGIN
11        S := S + I * I;
12        I := I + 1;
13    END;
14    CALL WRITEI(S);
15    CALL WRITELN;
16 END.
```

## 1.2 Mô tả chương trình

Chương trình thực hiện các bước sau:

1. Khởi tạo biến  $S = 0$  và  $I = 1$
2. Vòng lặp WHILE: Khi  $I \leq 5$ , thực hiện:
  - $S := S + I * I$  (cộng bình phương của  $I$  vào  $S$ )
  - $I := I + 1$  (tăng  $I$  lên 1)
3. In giá trị  $S$  ra màn hình
4. Xuống dòng

Kết quả mong đợi:  $S = 0 + 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$

## 1.3 Generated Code

Listing 2: Generated Code cho example1.kpl

```
1 0: J 1
2 1: INT 6
3 2: LA 0 ,4
4 3: LC 0
5 4: ST
6 5: LA 0 ,5
7 6: LC 1
8 7: ST
```

```

9| 8:  LV 0 ,5
10| 9:  LC 5
11| 10: LE
12| 11: FJ 25
13| 12: LA 0 ,4
14| 13: LV 0 ,4
15| 14: LV 0 ,5
16| 15: LV 0 ,5
17| 16: ML
18| 17: AD
19| 18: ST
20| 19: LA 0 ,5
21| 20: LV 0 ,5
22| 21: LC 1
23| 22: AD
24| 23: ST
25| 24: J 8
26| 25: LV 0 ,4
27| 26: WRI
28| 27: WLN
29| 28: HL

```

## 1.4 Phân tích chi tiết từng phần

### 1.4.1 Khởi tạo Stack Frame

<b>Inst</b>	<b>Code</b>	<b>Giải thích</b>
0	J 1	Jump đến instruction 1 (bỏ qua code của procedure declarations)
1	INT 6	Cấp phát stack frame với kích thước 6 words. <ul style="list-style-type: none"> <li>• Offset 0-3: Reserved cho return address, static link, dynamic link, etc.</li> <li>• Offset 4: Biến S (kiểu INTEGER)</li> <li>• Offset 5: Biến I (kiểu INTEGER)</li> </ul>

### 1.4.2 Khởi tạo biến S := 0

<b>Inst</b>	<b>Code</b>	<b>Giải thích</b>
2	LA 0 ,4	Load Address: Đẩy địa chỉ của biến S (offset 4, level 0) lên stack Stack: [addr_S]
3	LC 0	Load Constant: Đẩy hằng số 0 lên stack Stack: [addr_S, 0]
4	ST	Store: Lấy giá trị 0 từ đỉnh stack, lấy địa chỉ từ dưới, gán S = 0 Stack: [] (rỗng sau khi store)

### 1.4.3 Khởi tạo biến I := 1

Inst	Code	Giải thích
5	LA 0,5	Load Address: Đẩy địa chỉ của biến I (offset 5, level 0) lên stack Stack: [addr_I]
6	LC 1	Load Constant: Đẩy hằng số 1 lên stack Stack: [addr_I, 1]
7	ST	Store: Gán I = 1 Stack: []

### 1.4.4 Vòng lặp WHILE I <= 5

#### Điều kiện kiểm tra (Instructions 8-11)

Inst	Code	Giải thích
8	LV 0,5	Load Value: Đẩy giá trị của biến I lên stack Stack: [I]
9	LC 5	Load Constant: Đẩy hằng số 5 lên stack Stack: [I, 5]
10	LE	Less or Equal: So sánh I <= 5 Lấy 2 giá trị từ stack, so sánh, đẩy kết quả boolean lên stack Stack: [true/false]
11	FJ 25	False Jump: Nếu kết quả là false, nhảy đến instruction 25 (thoát vòng lặp) Nếu true, tiếp tục thực hiện thân vòng lặp

**Thân vòng lặp: S := S + I\*I (Instructions 12-18)** Đây là phần phức tạp nhất, cần tính toán biểu thức  $S + I*I$ . Thứ tự thực hiện:

1. Tính  $I*I$  trước (theo quy tắc precedence)
2. Sau đó cộng với  $S$
3. Cuối cùng gán vào  $S$

Inst	Code	Giải thích
12	LA 0,4	Load Address của S (để sau này store kết quả) Stack: [addr_S]
13	LV 0,4	Load Value của S (giá trị hiện tại của S) Stack: [addr_S, S]
14	LV 0,5	Load Value của I (tổng hạng thứ nhất cho phép nhân) Stack: [addr_S, S, I]

15	LV 0,5	Load Value của I (toán hạng thứ hai cho phép nhân) Stack: [addr_S, S, I, I]
16	ML	Multiply: Nhân I * I Lấy 2 giá trị cuối từ stack, nhân, đẩy kết quả lên Stack: [addr_S, S, I*I]
17	AD	Add: Cộng S + (I*I) Lấy 2 giá trị cuối từ stack, cộng, đẩy kết quả lên Stack: [addr_S, S+I*I]
18	ST	Store: Gán S = S + I*I Lấy giá trị S+I*I từ đỉnh stack, lấy địa chỉ addr_S từ dưới, gán Stack: []

**Lưu ý quan trọng:** Thứ tự các instruction phải đúng để stack operations hoạt động chính xác. Nếu thiếu bất kỳ LV nào, stack sẽ ở trạng thái sai và kết quả sẽ không đúng.

### Tăng biến I := I + 1 (Instructions 19-23)

Inst	Code	Giải thích
19	LA 0,5	Load Address của I Stack: [addr_I]
20	LV 0,5	Load Value của I (giá trị hiện tại) Stack: [addr_I, I]
21	LC 1	Load Constant 1 Stack: [addr_I, I, 1]
22	AD	Add: I + 1 Stack: [addr_I, I+1]
23	ST	Store: Gán I = I + 1 Stack: []

### Nhảy về đầu vòng lặp (Instruction 24)

Inst	Code	Giải thích
24	J 8	Jump: Nhảy vô điều kiện về instruction 8 (đầu vòng lặp) Kiểm tra lại điều kiện I <= 5

### 1.4.5 In kết quả (Instructions 25-27)

Inst	Code	Giải thích
25	LV 0,4	Load Value của S (để in ra) Stack: [S]

26	WRI	Write Integer: In giá trị integer từ đỉnh stack ra màn hình Đây là predefined procedure WRITEI(S) Stack: []
27	WLN	Write Line: Xuống dòng Đây là predefined procedure WRITELN

#### 1.4.6 Kết thúc chương trình (Instruction 28)

Inst	Code	Giải thích
28	HL	Halt: Dừng chương trình

## 2 Phân tích chi tiết Example 3 - Chương trình có chương trình con

### 2.1 Source Code

Listing 3: example3.kpl - Tower of Hanoi với PROCEDURE

```
1 PROGRAM EXAMPLE3; (* TOWER OF HANOI *)
2 VAR   I:INTEGER;
3     N:INTEGER;
4     P:INTEGER;
5     Q:INTEGER;
6     C:CHAR;
7
8 PROCEDURE HANOI(N:INTEGER; S:INTEGER; Z:INTEGER);
9 BEGIN
10   IF N != 0 THEN
11     BEGIN
12       CALL HANOI(N-1,S,6-S-Z);
13       I:=I+1;
14       CALL WRITELN;
15       CALL WRITEI(I);
16       CALL WRITEI(N);
17       CALL WRITEI(S);
18       CALL WRITEI(Z);
19       CALL HANOI(N-1,6-S-Z,Z)
20     END
21 END; (*END OF HANOI*)
22
23 BEGIN
24   FOR N := 1 TO 4 DO
25     BEGIN
26       FOR I:=1 TO 4 DO
27         CALL WRITEC(' ', ' ');
28       C := READC;
29       CALL WRITEC(C)
30     END;
31   P:=1;
32   Q:=2;
33   FOR N:=2 TO 4 DO
34     BEGIN
35       I:=0;
36       CALL HANOI(N,P,Q);
37       CALL WRITELN
38     END
39 END. (* TOWER OF HANOI *)
```

### 2.2 Mô tả chương trình

Chương trình bao gồm:

1. **PROCEDURE HANOI:** Hàm đệ quy giải bài toán Tower of Hanoi

- Nhận 3 tham số: N (số đĩa), S (cột nguồn), Z (cột đích)

- Nếu  $N \neq 0$ , gọi đệ quy và in thông tin

## 2. Main program:

- Vòng lặp FOR đọc 4 ký tự và in ra
- Khởi tạo  $P=1$ ,  $Q=2$
- Vòng lặp FOR gọi HANOI với  $N$  từ 2 đến 4

### 2.3 Generated Code

Listing 4: Generated Code cho example3.kpl

```

1 0:   J  41
2 1:   J  2
3 2:   INT 7
4 3:   HL
5 4:   LC  0
6 5:   NE
7 6:   FJ  41
8 7:   HL
9 8:   LC  1
10 9:   SB
11 10:  HL
12 11:  LC  6
13 12:  HL
14 13:  SB
15 14:  HL
16 15:  SB
17 16:  HL
18 17:  LA  0 ,4
19 18:  LV  0 ,4
20 19:  LC  1
21 20:  AD
22 21:  ST
23 22:  WLN
24 23:  LV  0 ,4
25 24:  WRI
26 25:  HL
27 26:  WRI
28 27:  HL
29 28:  WRI
30 29:  HL
31 30:  WRI
32 31:  HL
33 32:  LC  1
34 33:  SB
35 34:  LC  6
36 35:  HL
37 36:  SB
38 37:  HL
39 38:  SB
40 39:  HL
41 40:  HL
42 41:  INT 9
43 42:  LA  0 ,5
44 43:  LC  1

```

45	44: ST
46	45: LV 0,5
47	46: LC 4
48	47: LE
49	48: FJ 75
50	49: LA 0,4
51	50: LC 1
52	51: ST
53	52: LV 0,4
54	53: LC 4
55	54: LE
56	55: FJ 64
57	56: LC 32
58	57: WRC
59	58: LA 0,4
60	59: LV 0,4
61	60: LC 1
62	61: AD
63	62: ST
64	63: J 52
65	64: LA 0,8
66	65: RC
67	66: ST
68	67: LV 0,8
69	68: WRC
70	69: LA 0,5
71	70: LV 0,5
72	71: LC 1
73	72: AD
74	73: ST
75	74: J 45
76	75: LA 0,6
77	76: LC 1
78	77: ST
79	78: LA 0,7
80	79: LC 2
81	80: ST
82	81: LA 0,5
83	82: LC 2
84	83: ST
85	84: LV 0,5
86	85: LC 4
87	86: LE
88	87: FJ 102
89	88: LA 0,4
90	89: LC 0
91	90: ST
92	91: LV 0,5
93	92: LV 0,6
94	93: LV 0,7
95	94: HL
96	95: WLN
97	96: LA 0,5
98	97: LV 0,5
99	98: LC 1
100	99: AD
101	100: ST
102	101: J 84

## 2.4 Phân tích chi tiết

### 2.4.1 Phần PROCEDURE HANOI (Instructions 0-40)

Đây là phần code của PROCEDURE HANOI. Tuy nhiên, do Lab 7 - Code Gen 1 **chưa implement** việc gọi procedure và xử lý parameters, nên nhiều chỗ được thay thế bằng HL (halt).

Inst	Code	Giải thích
0	J 41	Jump đến main program (bỏ qua procedure code)
1	J 2	Jump đến body của procedure
2	INT 7	Cấp phát stack frame cho procedure (7 words)
3	HL	<b>CHƯA IMPLEMENT:</b> Load parameter N Trong Lab 7, chưa có cơ chế load parameters từ stack
4-5	LC 0, NE	So sánh N != 0 Tuy nhiên thiếu giá trị N (đã bị HL ở instruction 3)
6	FJ 41	Nếu false, exit procedure
7-16	HL (nhiều lần)	<b>CHƯA IMPLEMENT:</b> <ul style="list-style-type: none"> <li>Instruction 7: Load parameter S</li> <li>Instruction 10: Load parameter Z</li> <li>Instruction 12: Tính 6-S-Z</li> <li>Instruction 15: Gọi CALL HANOI(N-1, S, 6-S-Z)</li> </ul>
17-21	LA 0,4, LV 0,4, LC 1, AD, ST	<b>ĐÚNG:</b> I := I + 1
22	WLN	<b>ĐÚNG:</b> CALL WRITELN
23-24	LV 0,4, WRI	<b>ĐÚNG:</b> CALL WRITEI(I)
25-30	HL, WRI, HL, WRI, HL, WRI	<b>CHƯA IMPLEMENT:</b> <ul style="list-style-type: none"> <li>Thiếu load parameters N, S, Z</li> <li>CALL WRITEI(N), CALL WRITEI(S), CALL WRITEI(Z)</li> </ul>
31-39	HL (nhiều lần)	<b>CHƯA IMPLEMENT:</b> CALL HANOI(N-1, 6-S-Z, Z) - Gọi đệ quy

## 2.4.2 Phần Main Program (Instructions 41-102)

Phần main program được sinh mã **đầy đủ và chính xác**.

### Khởi tạo Stack Frame (Instruction 41)

Inst	Code	Giải thích
41	INT 9	Cấp phát stack frame với 9 words: <ul style="list-style-type: none"><li>• Offset 4: I (INTEGER)</li><li>• Offset 5: N (INTEGER)</li><li>• Offset 6: P (INTEGER)</li><li>• Offset 7: Q (INTEGER)</li><li>• Offset 8: C (CHAR)</li></ul>

### Vòng lặp FOR N := 1 TO 4 (Instructions 42-74)

Inst	Code	Giải thích
42-44	LA 0,5, LC 1, ST	Khởi tạo N := 1
45-48	LV 0,5, LC 4, LE, FJ 75	Điều kiện N <= 4
49-63	Vòng lặp FOR I := 1 TO 4 (nested loop) LA 0,4, LC 1, ST LV 0,4, LC 4, LE, FJ 64 LC 32, WRC LA 0,4, LV 0,4, LC 1, AD, ST J 52	I := 1 I <= 4 CALL WRITEC(' ') - in space I := I + 1 Jump về đầu vòng lặp I
64-68	LA 0,8, RC, ST, LV 0,8, WRC	<b>ĐÚNG:</b> <ul style="list-style-type: none"><li>• C := READC - đọc ký tự</li><li>• CALL WRITEC(C) - in ký tự</li></ul>
69-73	LA 0,5, LV 0,5, LC 1, AD, ST	Tăng N: N := N + 1
74	J 45	Jump về đầu vòng lặp N

### Khởi tạo P và Q (Instructions 75-80)

Inst	Code	Giải thích
75-77	LA 0,6, LC 1, ST	<b>ĐÚNG:</b> P := 1
78-80	LA 0,7, LC 2, ST	<b>ĐÚNG:</b> Q := 2

### Vòng lặp FOR N := 2 TO 4 (Instructions 81-101)

Inst	Code	Giải thích
81-83	LA 0,5, LC 2, ST	Khởi tạo N := 2
84-87	LV 0,5, LC 4, LE, FJ 102	Điều kiện N <= 4
88-90	LA 0,4, LC 0, ST	<b>ĐÚNG:</b> I := 0
91-94	LV 0,5, LV 0,6, LV 0,7, HL	<b>CHƯA IMPLEMENT:</b> CALL HANOI(N,P,Q) - Gọi procedure với parameters Cần load N, P, Q lên stack và gọi procedure
95	WLN	<b>ĐÚNG:</b> CALL WRITELN
96-100	LA 0,5, LV 0,5, LC 1, AD, ST	Tăng N: N := N + 1
101	J 84	Jump về đầu vòng lặp