





	A	B	C	D	E	F	G	H	I
1			business_overview						
2			사업개요						
3			사업내용	호텔건물 매입 및 운영 사업					
4			건물명	FS호텔					
5			주소	서울시 강남구 테헤란로					
6			건물규모	지하1층/지상10층					
7			객실수						
8			TypeA	40	객실				
9			TypeB	40	객실				
10			TypeC	20	객실				
11			자산매입						
12			자산매입금액	18,000,000,000	원				
13			매입부수비용	1,000,000,000	원				
14									
15			period_assumptions						
16			기본기간가정						
17			모델시작일	2023-12-01					
18			모델종료일	2027-01-31					
19			운영시작일	2024-01-01					
20			운영종료일	2026-12-31					
21			자산매입일정						
22			자산매입일	2023-12-31					
23			매입부수비용지급일	2023-12-31					
24			자금조달일정						
25			자기자본유입일	2023-12-31					
26			차입금유입일	2023-12-31					
27			이자지급시작일	2024-01-01					
28			이자지급종료일	2026-12-31					
29			원금상환시작일	2024-01-01					
30			원금상환종료일	2026-12-31					
31			연간인상률						
32			판매단가	5.0%					
33			운영비	3.0%					
34			인건비	5.0%					
35									
36			funding_assumptions						
37			equity						
38			amount	10,000,000,000	원				
39			최초배당일	2025-12-31					
40			배당률	30.0%	매년말 잔여 현금흐름 기준				
41			loan						
42			amount	10,000,000,000	원				
43			원금상환주기	3	개월				
44			상환원금	50,000,000	원				
45			이자지급주기	1	개월				
46			이자율	5.0%					
47									
48			facility_cost						
49			통상수선비						
50			TypeA	1,000	원				
51			TypeB	1,200	원				
52			TypeC	2,000	원				
53			대수선공사비						
54			TypeA	30,000,000	원				
55			TypeB	35,000,000	원				
56			TypeC	50,000,000	원				
57			수선시작일						
58			TypeA	2025-03-01					
59			TypeB	2025-04-01					
60			TypeC	2025-05-01					
61			수선종료일						
62			TypeA	2025-03-31					
63			TypeB	2025-04-30					
64			TypeC	2025-05-31					
65									
66			monthly_cash_flow						

Sheet1

## [가정 데이터 로딩 코드 예시]

```
wb = load_workbook("FCHotel_FSmodeling_assumption.xlsx")
ws = wb['Sheet1']

assumption = {}
for row in ws.iter_rows(values_only=True):
    if row[0] is not None:
        assumption[row[0]] = {}
        astndct1 = assumption[row[0]]
    elif row[1] is not None:
        astndct1[row[1]] = {}
        astndct2 = astndct1[row[1]]
    elif row[2] is not None:
        astndct2[row[2]] = row[3]

monthly_occ_rate = pd.DataFrame(assumption['monthly_occ_rate'])
room_operating_cost = pd.DataFrame(assumption['room_operating_cost'])
salary_cost = pd.DataFrame(assumption['salary_cost'])
```

Python

## 2-2. 기간 Index 설정 코드 작성

- 전체 사업기간, 운영기간, 주요 이벤트에 대한 기간 등을 index 변수에 설정

## [index 변수에 주요 기간 데이터 설정 예시]

```

index = {}

index["model"] = list_end_of_months(
    assumption['period_assumptions']['기본기간가정']['모델시작일'],
    assumption['period_assumptions']['기본기간가정']['모델종료일']
)

index["operating"] = list_end_of_months(
    assumption['period_assumptions']['기본기간가정']['운영시작일'],
    assumption['period_assumptions']['기본기간가정']['운영종료일']
)

index["이자지급"] = list_end_of_months(
    assumption['period_assumptions']['자금조달일정']['이자지급시작일'],
    assumption['period_assumptions']['자금조달일정']['이자지급종료일']
)

index["원금상환"] = list_end_of_months(
    assumption['period_assumptions']['자금조달일정']['원금상환시작일'],
    assumption['period_assumptions']['자금조달일정']['원금상환종료일']
)

index["수선TypeA"] = list_end_of_months(
    assumption['facility_cost']['수선시작일']['TypeA'],
    assumption['facility_cost']['수선종료일']['TypeA']
)

index["수선TypeB"] = list_end_of_months(
    assumption['facility_cost']['수선시작일']['TypeB'],
    assumption['facility_cost']['수선종료일']['TypeB']
)

index["수선TypeC"] = list_end_of_months(
    assumption['facility_cost']['수선시작일']['TypeC'],
    assumption['facility_cost']['수선종료일']['TypeC']
)

```

Python

## 2-3. 자금의 조달 및 소요에 관한 사항 설정 코드 작성

- 자기자본, 차입금 등 자금의 조달에 대해 설정하는 코드 작성
- 조달된 자금의 사용에 대해 설정하는 코드 작성

[자금의 조달/소요 설정 코드 예시]

### 3-1. 자기자본

```
data = []
for dt in index['model']:
    dct = {}
    if dt == assumption['period_assumptions']['자금조달일정']['자기자본유입일']:
        dct['자기자본유입'] = assumption['funding_assumptions']['equity']['amount']
    else:
        dct['자기자본유입'] = 0

    dct['배당금지급'] = 0

    data.append(dct)
funding['자기자본'] = pd.DataFrame(data, index=index['model'])
```

Python

### 3-2. 차입금

```
차입금잔액 = 0
data = []
for dt in index['model']:
    dct = {}

    if dt == assumption['period_assumptions']['자금조달일정']['차입금유입일']:
        차입금유입 = assumption['funding_assumptions']['loan']['amount']
    else:
        차입금유입 = 0

    if (dt in index['원금상환']) and (index['원금상환'].index(dt) % 3 == 2):
        차입금상환 = assumption['funding_assumptions']['loan']['상환원금']
    else:
        차입금상환 = 0

    if dt in index['이자지급']:
        이자율 = assumption['funding_assumptions']['loan']['이자율']
        월간일수 = index['days'].loc[dt, '월간일수']
        연간일수 = index['days'].loc[dt, '연간일수']
        차입이자 = int(round(차입금잔액 * 이자율 * 월간일수 / 연간일수, -1))
    else:
        차입이자 = 0

    차입금잔액 = 차입금잔액 + 차입금유입 - 차입금상환

    dct['차입금유입'] = 차입금유입
    dct['차입금상환'] = 차입금상환
    dct['차입이자'] = 차입이자
    dct['차입금잔액'] = 차입금잔액

    data.append(dct)
funding['차입금'] = pd.DataFrame(data, index=index['model'])
```

Python

## 2-4. 운영수입 및 운영비용에 관한 사항 코드 작성

- model index에 따라 해당 기간에 발생하는 운영수입과 운영비용에 대한 인식 방법을 코드로 작성
- 운영수입의 경우 총객실수, 판매가능객실수 등 운영수입 계산을 위한 기초 계산 완료 후 최종적으로 해당 기간에 인식되는 객실수입을 계산
- 운영비용의 경우 운영수입에 연동되어 발생하는 비용은 운영수입에 연동되도록 코드 작성하며, 고정비는 주어진 가정에 맞춰서 코드 작성함

### [운영수입 계산 코드 예시]

```
room_type_list = list(assumption['business_overview']['객실수'].keys())
for room_type in room_type_list:
    data = []
    for dt in index['model']:
        dct = {}

        if dt in index['operating']:
            # 총객실수
            dct['총객실수'] = (
                assumption['business_overview']['객실수'][room_type] *
                index['days'].loc[dt, '월간일수']
            )

            # 사용불가객실수(대수선 기간)
            if dt in index['수선' + room_type]:
                dct['사용불가객실수'] = (
                    assumption['business_overview']['객실수'][room_type] *
                    index['days'].loc[dt, '월간일수']
                )
            else:
                dct['사용불가객실수'] = 0

            # 판매가능객실수
            dct['판매가능객실수'] = dct['총객실수'] - dct['사용불가객실수']

            # 객실판매비율(OCC rate)
            dct['객실판매비율'] = assumption['monthly_occ_rate'][room_type][dt.month]

            # 판매객실수
            dct['판매객실수'] = round(dct['판매가능객실수'] * dct['객실판매비율'])

            # 판매단가
            dct['판매단가'] = assumption['monthly_price'][room_type][dt.month]

            # 객실수입
            dct['객실수입'] = dct['판매객실수'] * dct['판매단가']
        else:
            dct['총객실수'] = 0
            dct['사용불가객실수'] = 0
            dct['판매가능객실수'] = 0
            dct['객실판매비율'] = 0.0
            dct['판매객실수'] = 0
            dct['판매단가'] = 0
            dct['객실수입'] = 0

        data.append(dct)
    operating_income[room_type] = pd.DataFrame(data, index=index['model'])
```

Python

## [운영비용 계산 코드 예시]

## 5-1-1. 청소세탁비

```

data = []
for dt in index['model']:
    dct = {}

    if dt in index['operating']:
        dct['TypeA'] = int(
            operating_income['TypeA'].loc[dt, '판매객실수'] *
            assumption['room_operating_cost']['청소세탁비']['TypeA'] *
            index['연간인상률'].loc[dt, '운영비']
        )
        dct['TypeB'] = int(
            operating_income['TypeB'].loc[dt, '판매객실수'] *
            assumption['room_operating_cost']['청소세탁비']['TypeB'] *
            index['연간인상률'].loc[dt, '운영비']
        )
        dct['TypeC'] = int(
            operating_income['TypeC'].loc[dt, '판매객실수'] *
            assumption['room_operating_cost']['청소세탁비']['TypeC'] *
            index['연간인상률'].loc[dt, '운영비']
        )
    else:
        dct['TypeA'] = 0
        dct['TypeB'] = 0
        dct['TypeC'] = 0

    dct['Total'] = dct['TypeA'] + dct['TypeB'] + dct['TypeC']
    data.append(dct)

operating_cost['객실운영비']['청소세탁비'] = pd.DataFrame(data, index=index['model'])

```

Python

## 5-1-2. 수도광열비

```

data = []
for dt in index['model']:
    dct = {}

    if dt in index['operating']:
        dct['TypeA'] = int(
            operating_income['TypeA'].loc[dt, '판매객실수'] *
            assumption['room_operating_cost']['수도광열비']['TypeA'] *
            index['연간인상률'].loc[dt, '운영비']
        )
        dct['TypeB'] = int(
            operating_income['TypeB'].loc[dt, '판매객실수'] *
            assumption['room_operating_cost']['수도광열비']['TypeB'] *
            index['연간인상률'].loc[dt, '운영비']
        )
        dct['TypeC'] = int(
            operating_income['TypeC'].loc[dt, '판매객실수'] *
            assumption['room_operating_cost']['수도광열비']['TypeC'] *

```



## 2-5. 현금흐름 계산

- model index에 따라 현금의 유입과 유출을 기록
- 유입과 유출의 history는 "cashflow" 변수에 저장
- 기초/기말현금, 입금금액, 출금금액은 "balance" 변수에 저장

[현금흐름 기록 코드 예시]

```

idx = 0
cash_balance = 0
for dt in index['model']:
    ##### 0. 기초현금 계산
    balance.loc[dt, '기초현금'] = cash_balance

    ##### 1. 자금조달소요
    ## 1-1. 자기자본 유입
    amount = funding['자기자본'].loc[dt, '자기자본유입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자금조달', '자기자본', '자기자본유입', amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

    ## 1-2. 차입금 유입
    amount = funding['차입금'].loc[dt, '차입금유입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자금조달', '차입금', '차입금유입', amount, 0]
        balance.loc[dt, '입금금액'] += amount
        idx += 1

    ## 1-3. 자산매입
    amount = funding['자산매입'].loc[dt, '자산매입']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자산매입', '자산매입', '매입대금지출', 0, amount]
        balance.loc[dt, '출금금액'] += amount
        idx += 1
    amount = funding['자산매입'].loc[dt, '매입부수비용']
    if amount > 0:
        cashflow.loc[idx] = [dt, '자산매입', '매입부수비용', '부수비용지출', 0, amount]
        balance.loc[dt, '출금금액'] += amount
        idx += 1

    ##### 2. 운영수입
    for room_type in room_type_list:
        amount = operating_income[room_type].loc[dt, '객실수입']
        if amount > 0:
            cashflow.loc[idx] = [dt, '운영수입', '객실수입', room_type, amount, 0]
            balance.loc[dt, '입금금액'] += amount
            idx += 1

    ##### 3. 운영비용 - 객실운영비
    ## 3-1. 청소세탁비
    for room_type in room_type_list:
        amount = operating_cost['객실운영비']['청소세탁비'].loc[dt, room_type]
        if amount > 0:
            cashflow.loc[idx] = [dt, '객실운영비', '청소세탁비', room_type, 0, amount]
            balance.loc[dt, '출금금액'] += amount
            idx += 1

    ## 3-2. 수도광열비
    for room_type in room_type_list:
        amount = operating_cost['객실운영비']['수도광열비'].loc[dt, room_type]
        if amount > 0:
            cashflow.loc[idx] = [dt, '객실운영비', '수도광열비', room_type, 0, amount]
            balance.loc[dt, '출금금액'] += amount
            idx += 1

```

[현금흐름 최종 결과물 예시]

categoryA	기초현금	자금조달		자산매입		운영수입			
categoryB		자기자본	차입금	자산매입	매입부수비용	객실수입			소계
categoryC		자기자본유입	차입금유입	매입대금지출	부수비용지출	TypeA	TypeB	TypeC	
date									
2023-12-31	0	10000000000	10000000000	-18000000000	-10000000000	0	0	0	0
2024-01-31	10000000000	0	0	0	0	145080000	167400000	131750000	444230000
2024-02-29	1238417103	0	0	0	0	128180000	147900000	116000000	392080000
2024-03-31	1438862561	0	0	0	0	99200000	119040000	86800000	305040000
2024-04-30	1519606164	0	0	0	0	96000000	115200000	84000000	295200000
2024-05-31	1645595214	0	0	0	0	99200000	119040000	93000000	311240000
2024-06-30	1781572567	0	0	0	0	84000000	100800000	84000000	268800000
2024-07-31	1836177617	0	0	0	0	145080000	167400000	131750000	444230000
2024-08-31	2085018220	0	0	0	0	145080000	167400000	131750000	444230000
2024-09-30	2333858823	0	0	0	0	96000000	115200000	90000000	301200000
2024-10-31	2414912793	0	0	0	0	99200000	119040000	93000000	311240000
2024-11-30	2551313646	0	0	0	0	84000000	100800000	84000000	268800000
2024-12-31	2656328536	0	0	0	0	145080000	167400000	139500000	451980000
2025-01-31	2851906389	0	0	0	0	145080000	167400000	131750000	444230000
2025-02-28	3085386233	0	0	0	0	123760000	142800000	112000000	378560000
2025-03-31	3272938740	0	0	0	0	0	119040000	86800000	205840000
2025-04-30	2069689468	0	0	0	0	96000000	0	84000000	180000000

[엑셀파일 출력 결과물 예시]

	A	B	C	D	E	F	G	H	I	J
1	categoryA	기초현금	자금조달		자산매입		운영수입			
2	categoryB		자기자본	차입금	자산매입	매입부수비용	객실수입			소계
3	categoryC		자기자본유입	차입금유입	매입대금지출	부수비용지출	TypeA	TypeB	TypeC	
4	date									
5	2023-12-31	0	10,000,000,000	10,000,000,000	-18,000,000,000	-1,000,000,000	0	0	0	0
6	2024-01-31	905,871,236	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
7	2024-02-29	1,144,288,339	0	0	0	0	128,180,000	147,900,000	116,000,000	392,080,000
8	2024-03-31	1,344,733,797	0	0	0	0	99,200,000	119,040,000	86,800,000	305,040,000
9	2024-04-30	1,425,477,400	0	0	0	0	96,000,000	115,200,000	84,000,000	295,200,000
10	2024-05-31	1,551,466,450	0	0	0	0	99,200,000	119,040,000	93,000,000	311,240,000
11	2024-06-30	1,687,443,803	0	0	0	0	84,000,000	100,800,000	84,000,000	268,800,000
12	2024-07-31	1,742,048,853	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
13	2024-08-31	1,990,889,456	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
14	2024-09-30	2,239,730,059	0	0	0	0	96,000,000	115,200,000	90,000,000	301,200,000
15	2024-10-31	2,320,784,029	0	0	0	0	99,200,000	119,040,000	93,000,000	311,240,000
16	2024-11-30	2,457,184,882	0	0	0	0	84,000,000	100,800,000	84,000,000	268,800,000
17	2024-12-31	2,562,199,772	0	0	0	0	145,080,000	167,400,000	139,500,000	451,980,000
18	2025-01-31	2,757,777,625	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
19	2025-02-28	2,991,257,469	0	0	0	0	123,760,000	142,800,000	112,000,000	378,560,000
20	2025-03-31	3,178,809,976	0	0	0	0	0	119,040,000	86,800,000	205,840,000
21	2025-04-30	1,975,560,704	0	0	0	0	96,000,000	0	84,000,000	180,000,000
22	2025-05-31	604,784,793	0	0	0	0	99,200,000	119,040,000	0	218,240,000
23	2025-06-30	-338,267,669	0	0	0	0	84,000,000	100,800,000	84,000,000	268,800,000
24	2025-07-31	-287,541,820	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
25	2025-08-31	-43,337,316	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
26	2025-09-30	200,867,188	0	0	0	0	96,000,000	115,200,000	90,000,000	301,200,000
27	2025-10-31	277,886,997	0	0	0	0	99,200,000	119,040,000	93,000,000	311,240,000
28	2025-11-30	410,142,795	0	0	0	0	84,000,000	100,800,000	84,000,000	268,800,000
29	2025-12-31	511,279,604	0	0	0	0	145,080,000	167,400,000	139,500,000	451,980,000
30	2026-01-31	701,892,178	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
31	2026-02-28	930,473,433	0	0	0	0	123,760,000	142,800,000	112,000,000	378,560,000
32	2026-03-31	1,113,561,418	0	0	0	0	99,200,000	119,040,000	86,800,000	305,040,000
33	2026-04-30	1,186,134,228	0	0	0	0	96,000,000	115,200,000	84,000,000	295,200,000
34	2026-05-31	1,304,171,427	0	0	0	0	99,200,000	119,040,000	93,000,000	311,240,000
35	2026-06-30	1,431,918,154	0	0	0	0	84,000,000	100,800,000	84,000,000	268,800,000
36	2026-07-31	1,478,828,594	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
37	2026-08-31	1,718,443,499	0	0	0	0	145,080,000	167,400,000	131,750,000	444,230,000
38	2026-09-30	1,958,058,404	0	0	0	0	96,000,000	115,200,000	90,000,000	301,200,000
39	2026-10-31	2,031,102,619	0	0	0	0	99,200,000	119,040,000	93,000,000	311,240,000
40	2026-11-30	2,159,273,996	0	0	0	0	84,000,000	100,800,000	84,000,000	268,800,000
41	2026-12-31	2,256,595,396	0	0	0	0	145,080,000	167,400,000	139,500,000	451,980,000
42	2027-01-31	2,442,278,718	0	0	0	0	0	0	0	0
43										

Sheet1

+