

9week

활용3-(5); 조건에 맞는 형태의 파일 만들기

‘ cf. 활용3-(4); 길이 다른 파일 붙이기

‘ [조건]

‘ file1: 10줄, file2: 5줄

‘ Q1. file1 – file2(위)

‘ Q2. file2(위) - file1

‘ Q3. file1 – file2(아래)

‘ Q4. file2(아래) - file1

#1; For문

Sub k() ‘ **test.txt** 파일 생성

Open "C:\Users\USER\For 통특\test.txt" For Output As #1

For i = 1 To 9

Print #1, i

Next i

Close #1

End Sub

Sub kk() ‘ **test.txt** 파일 읽기

Open "C:\Users\USER\For 통특\test.txt" For Input As #1

For i = 1 To 5 ‘ **1~5번째 줄까지 읽음**

Line Input #1, a

Debug.Print a

Next i

Debug.Print "-----"

For i = 1 To 2 ‘ **이어서 6~7번째 줄 읽음**

Line Input #1, a

Debug.Print a

Next i

Debug.Print "-----"

For i = 1 To 2 ‘ **나머지 8~9번째 줄 읽음**

Line Input #1, a

Debug.Print a

Next i

Close #1

End Sub

#2; While문

Sub kk()

Open "C:\Users\USER\For 통특\test.txt" For Input As #1

While Not EOF(1) ‘ **파일의 끝까지 읽음**

Line Input #1, b

Debug.Print b

Wend

```

Debug.Print "-----"
While Not EOF(1)
    Line Input #1, b ' 이미 끝까지 읽었으므로 더 이상 읽을 게 X
    Debug.Print b
Wend
Close #1
End Sub

```

```

Sub kk()
    Open "C:\Users\USER\For 통특\test.txt" For Input As #1
    While Not EOF(1) ' if문을 이용해서 원하는 곳까지 읽기
        Line Input #1, b
        Debug.Print b
        cnt = cnt + 1
        If cnt = 5 Then End ' end; 종료
        If cnt = 5 Then GoTo 10 ' 10:으로 가기; 결국 종료
    Wend
    Debug.Print cnt
10:
End
End Sub

```

#3; Q1

```

Sub kk()
    Open "C:\Users\USER\For 통특\q1.txt" For Input As #1
    Open "C:\Users\USER\For 통특\q2.txt" For Input As #2
    While Not EOF(2) ' 짧은 파일로 읽기; 오류 방지를 위해
        Line Input #1, q1_long
        Line Input #2, q2_short
        Debug.Print q1_long; ","; q2_short ' 세미콜론(;); 잇따라서
    Wend
    While Not EOF(1) ' 긴 파일을 마저 읽기
        Line Input #1, q1_long
        Debug.Print q1_long
    Wend
End
End Sub

```

#4; Q2

```

Sub kk()
    Open "C:\Users\USER\For 통특\q1.txt" For Input As #1
    Open "C:\Users\USER\For 통특\q2.txt" For Input As #2
    While Not EOF(2) ' 역시 짧은 파일로 돌리기
        Line Input #1, q1_long
        Line Input #2, q2_short
        Debug.Print q2_short; ","; q1_long
    Wend

```

```

While Not EOF(1) ' 남은 긴 파일 읽기
    Line Input #1, q1_long
    'Debug.Print "   "; q1_long ' 공백으로 구분
    Debug.Print ","; q1_long ' 구분자로 구분(추천)
Wend
End
End Sub

```

#5; Q3

```

Sub kk()
    Open "C:\Users\USER\For 통특\q1.txt" For Input As #1
    Open "C:\Users\USER\For 통특\q2.txt" For Input As #2
    For i = 1 To 5 ' 긴 파일의 5줄을 먼저 읽기
        Line Input #1, q1_long
        Debug.Print q1_long
    Next i
    While Not EOF(1) ' 긴 파일의 나머지 읽기
        Line Input #1, q1_long
        Line Input #2, q2_short
        Debug.Print q1_long; ", "; q2_short ' q1, q2 잇따라서
    Wend
End
End Sub

```

#6; Q4

```

Sub kk()
    Open "C:\Users\USER\For 통특\q1.txt" For Input As #1
    Open "C:\Users\USER\For 통특\q2.txt" For Input As #2
    For i = 1 To 5 ' 긴 파일의 5줄을 먼저 읽기 + 앞부분 공백
        Line Input #1, q1_long
        Debug.Print String(4, " "); q1_long 'string; 특정 문자열 반복
    Next i
    While Not EOF(1) ' 긴 파일의 나머지 읽기
        Line Input #1, q1_long
        Line Input #2, q2_short
        Debug.Print q2_short; ", "; q1_long ' q2, q1 잇따라서
    Wend
End
End Sub

```

활용4-(1); 원하는 데이터 출력하기: 평균값 추출

#1; 일렬로 나열된 경우: 동일한 위치

```

Sub kk()
    Open "C:\Users\USER\For 통특\qq1.txt" For Input As #1
    While Not EOF(1) ' 파일의 끝을 모르니까

```

```

Line Input #1, in_str
For i = 1 To Len(in_str) ' 줄의 길이가 모두 다르니까
    If Mid(in_str, i, 4) = "mean" Then
        Debug.Print Mid(in_str, i + 5, 2) ' i에서 5칸 더 가서 2개를 뽑아라
    End If
Next i
Wend
End
End Sub

```

#2; 대소문자 혼합

```

Sub cf()
    x = "abcDEF"
    Debug.Print UCase(x) ' UCase(); 대문자로 바꿔주는 함수
    Debug.Print LCase(x) ' LCase(); 소문자로 바꿔주는 함수
End Sub

```

```

Sub kk()
    Open "C:\Users\USER\For 통특\qq2.txt" For Input As #1
    While Not EOF(1)
        Line Input #1, in_str
        For i = 1 To Len(in_str)
            If UCase(Mid(in_str, i, 4)) = "MEAN" Then ' UCase(or LCase)로 문자 통일
                Debug.Print Mid(in_str, i + 5, 2)
            End If
        Next i
    Wend
End
End Sub

```

#3; 대소문자 혼합 및 위치 다름

- ' #2와 동일
- ' 어차피 빈 공간은 건너뛰니까 의미 없음

```

Sub kk()
    Open "C:\Users\USER\For 통특\qq3.txt" For Input As #1
    While Not EOF(1)
        Line Input #1, in_str
        For i = 1 To Len(in_str)
            If UCase(Mid(in_str, i, 4)) = "MEAN" Then
                Debug.Print Mid(in_str, i + 5, 2)
            End If
        Next i
    Wend
End
End Sub

```


#1; Sorting X, 배열 X – 파일 2개

Sub kk1() ' 모든 경우의 수를 다 비교하기; 비효율적

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1

' Timer; 시간을 재는 함수

st = Timer ' start time

While Not EOF(1) ' 먼저 1번 파일을 열고

Input #1, num_tt1

Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2

While Not EOF(2) ' 2번 파일에 대해서

Input #2, num_tt2

If num_tt1 = num_tt2 Then Debug.Print num_tt1 ' 1번 파일의 값과 같은 지 확인

Wend

Close #2

Wend

et = Timer ' end time

Debug.Print "소요시간1: "; et - st ' 소요시간 출력

End

End Sub

#2; Sorting O, 배열 X – 파일 2개

Sub kk2() ' GoTo

st = Timer

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1

While Not EOF(1)

Input #1, num_tt1

Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2

While Not EOF(2)

Input #2, num_tt2

If num_tt1 = num_tt2 Then

Debug.Print num_tt1

GoTo 10: ' 10:이 있는 곳으로 가라

End If

Wend

10: Close #2

Wend

et = Timer

Debug.Print "소요시간2: "; et - st ' 아까보다 빨라짐

End

End Sub

Sub kk3() ' Select

st = Timer

```

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1
Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2
Input #1, num_tt1
Input #2, num_tt2
While Not EOF(1)
    Select Case num_tt1
        Case Is < num_tt2
            Input #1, num_tt1
        Case Is > num_tt2
            Input #2, num_tt2
        Case Is = num_tt2
            Debug.Print num_tt1 ' or num_tt2
            Input #1, num_tt1 ' 그 다음 이어서 읽기
    End Select
Wend ' 9 출력X; 파일의 끝까지 읽어서 While문 종료
While Not EOF(2)
    Select Case num_tt1
        Case Is < num_tt2 ' 더 이상 비교할 필요 없음
            GoTo 10: ' 나가기
        Case Is > num_tt2
            Input #2, num_tt2 ' 다음 tt2로 넘어가기
        Case Is = num_tt2
            Debug.Print num_tt2 ' 같으면 출력하고 종료
            GoTo 10:
    End Select
Wend
10:
et = Timer
Debug.Print "소요시간3:"; et - st ' 가장 효율적
End
End Sub

```

10week

활용4-(4); 공통된 ID 추출하기

- ' 1) 자료가 sorting 되어있지 않고, 배열 선언이 되어있지 않은 경우: 가장 비효율적
- ' 2) 자료가 sorting 되어있고, 배열 선언이 되어있지 않은 경우: GoTo
- ' 3) 자료가 sorting 되어있거나 되어있지 않고, 배열 선언이 되어있는 경우: 최대값으로 배열 잡기

#1; 최대값 구하기 for 배열이 크기를 잡기 위해

```

Sub kk4()
    Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1
    Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2
    ' 파일의 개수가 n개일 때, 방 안의 개수가 n이다 = 공통으로 존재 한다
    ' tt1에서의 최대값:9, 최소값:0
    ' tt2에서의 최대값:100, 최소값:0
    max_num = 0 ' 최대값 for 방의 크기를 정하기 위해서

```

```

min_num = 9999999
While Not EOF(1) ' 2번 파일에 대해서도 동일하게
    Input #1, num
    If max_num < num Then max_num = num
    If min_num > num Then min_num = num
Wend
Debug.Print max_num, min_num
End
End Sub

```

#2; Sorting O/X, 배열 O – 파일 2개

```

Const max_num = 100000 ' Const; 상수 선언
Dim cnt(max_num) As Integer ' 최대값만큼 배열 잡기
Sub kk4()
    st = Timer
    Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1
    Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2
' tt1, tt2의 최대값:9, 100000
    While Not EOF(1)
        Input #1, num_tt1
        cnt(num_tt1) = cnt(num_tt1) + 1
    Wend
    While Not EOF(2)
        Input #2, num_tt2
        cnt(num_tt2) = cnt(num_tt2) + 1 ' 방 안의 데이터가 2개다 = 두 파일에 모두 존재
    Wend
    For i = 1 To max_num
        If cnt(i) = 2 Then Debug.Print i, cnt(i) ' 방 번호(=공통된 ID)와 방 안의 개수(=파일 개수) 출력
    Next i
    et = Timer
    Debug.Print "소요시간4:"; et - st ' 배열은 은근 메모리를 잡아먹음
End
End Sub

```

#3; Sorting X, 배열 X – 파일 3개

Sub m1() ' sort X, 배열 X; 가장 비효율

```

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1
Open "C:\Users\USER\For 통특\Wtt12.txt" For Output As #3 ' tt1과 tt2의 공통된 데이터 저장

While Not EOF(1)
    Input #1, num_tt1
    Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2 ' 처음부터 다시 읽기 위해서
    While Not EOF(2)
        Input #2, num_Tt2
        If num_tt1 = num_Tt2 Then Print #3, num_tt1
    Wend

```



```

    Close #2
Wend
Close #1, #2, #3

Open "C:\Users\WUSER\For 통특\Wtt12.txt" For Input As #1
While Not EOF(1)
    Input #1, num_tt12
    Open "C:\Users\WUSER\For 통특\Wtt3.txt" For Input As #2
    While Not EOF(2)
        Input #2, num_tt3
        If num_tt12 = num_tt3 Then Debug.Print num_tt3
    Wend
    Close #2
Wend
End
End Sub

```

#4: Sorting O, 배열 X – 파일 3개

```

Sub m2() ' sort O, 배열 X; GoTo

    Open "C:\Users\WUSER\For 통특\Wtt1.txt" For Input As #1
    Open "C:\Users\WUSER\For 통특\Wtt12.txt" For Output As #3
    While Not EOF(1)
        Input #1, num_tt1
        Open "C:\Users\WUSER\For 통특\Wtt2.txt" For Input As #2
        While Not EOF(2)
            Input #2, num_tt2
            If num_tt1 = num_tt2 Then
                Print #3, num_tt1
                GoTo 10:
            End If
        Wend
    10: Close #2
    Wend
    Close #1, #2, #3

    Open "C:\Users\WUSER\For 통특\Wtt12.txt" For Input As #1
    While Not EOF(1)
        Input #1, num_tt12
        Open "C:\Users\WUSER\For 통특\Wtt3.txt" For Input As #2
        While Not EOF(2)
            Input #2, num_tt3
            If num_tt12 = num_tt3 Then
                Debug.Print num_tt3
                GoTo 30:
            End If
        Wend
    30: Close #2

```

Wend

End

End Sub

Sub m3() ' **sort O, 배열 X; Select**

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1

Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2

Open "C:\Users\USER\For 통특\Wtt12.txt" For Output As #3

Input #1, num_tt1

Input #2, num_tt2

While Not EOF(1)

Select Case num_tt1

Case Is < num_tt2

Input #1, num_tt1

Case Is > num_tt2

Input #2, num_tt2

Case Is = num_tt2

Print #3, num_tt1

Input #1, num_tt1

End Select

Wend ' **마지막이 저장이 안 되는 문제**

While Not EOF(2)

Select Case num_tt1

Case Is < num_tt2

GoTo 10:

Case Is > num_tt2

Input #2, num_tt2

Case Is = num_tt2

Print #3, num_tt1

GoTo 10:

End Select

Wend

10: Close #1, #2, #3

Open "C:\Users\USER\For 통특\Wtt12.txt" For Input As #1

Open "C:\Users\USER\For 통특\Wtt3.txt" For Input As #2

Input #1, num_tt12

Input #2, num_tt3

While Not EOF(1)

Select Case num_tt12

Case Is < num_tt3

Input #1, num_tt12

Case Is > num_tt3

Input #2, num_tt3

Case Is = num_tt3

Debug.Print num_tt3

Input #1, num_tt12

```

End Select
Wend
While Not EOF(2)
Select Case num_tt12
Case Is < num_tt3
GoTo 20:
Case Is > num_tt3
Input #2, num_tt3
Case Is = num_tt3
Debug.Print num_tt12
GoTo 20:
End Select
Wend
20:
End
End Sub

```

#5; Sorting O/X, 배열 O – 파일 3개

Sub m4() ' **sort O/X, 배열 O**

' (1) 방을 잡기위한 최대값 구하기; tt1:9, tt2:1000000, tt3:20

```

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1
Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2
Open "C:\Users\USER\For 통특\Wtt3.txt" For Input As #3
max_num = 0
While Not EOF(1) ' 나머지 파일에도 동일하게 적용해서 최대값 구하기
Input #1, num
If max_num < num Then max_num = num
Wend
Debug.Print max_num

```

' (2) 배열을 잡고 방 안의 개수를 이용해 공통된 ID 추출

```

Open "C:\Users\USER\For 통특\Wtt1.txt" For Input As #1
Open "C:\Users\USER\For 통특\Wtt2.txt" For Input As #2
Open "C:\Users\USER\For 통특\Wtt3.txt" For Input As #3
Dim cnt(1000000) As Integer ' As Integer; 메모리 절약을 위해
While Not EOF(1)
Input #1, num_tt1
cnt(num_tt1) = cnt(num_tt1) + 1
Wend
While Not EOF(2)
Input #2, num_tt2
cnt(num_tt2) = cnt(num_tt2) + 1
Wend
While Not EOF(3)
Input #3, num_tt3
cnt(num_tt3) = cnt(num_tt3) + 1
Wend

```

```
For i = 1 To 1000000
    If cnt(i) = 3 Then Debug.Print i, cnt(i)
Next i

End
End Sub
```

활용5-(1); ID에 따라 자료 분리

#1; 파일 생성 후 고객 별로 자료 분리 (복습)

```
Sub kk()

' (1) 먼저 파일 생성
Open "C:\Users\USER\For 통특\Wex.txt" For Output As #1 '
For i = 1 To 1000
    Randomize ' 초기화
    a = Int(Rnd * 10 + 1) ' 1~10 난수; 고객의 ID
    b = i ' 1-1000
    c = Int(Rnd * 2 + 1) ' 1~3 난수
    Print #1, a, b, c
Next i
Close #1
```

' (2) 고객 별로 자료 분리하기

```
Open "C:\Users\USER\For 통특\Wex.txt" For Input As #1
While Not EOF(1)
    Input #1, a, b, c
    For i = 1 To 10 ' cf. Output; 기존 자료 없어짐
        Open "C:\Users\USER\For 통특\Wu" + Trim(Str(a)) + ".txt" For Append As #2 ' Append; 자료 추가
        Print #2, a, b, c
        Close #2 ' 닫아야지 while문을 돌 때 오류 X
    Next i
Wend

End
End Sub
```

활용5-(2); 8:2 나누기

#1; 원하는 비율로 랜덤하게 파일 나누기

```
Sub kk()

' (1) 먼저 파일이 몇 줄인지 파악하기
Open "C:\Users\USER\For 통특\Wex.txt" For Input As #1
While Not EOF(1)
    Line Input #1, in_str
    cnt = cnt + 1 ' 파일의 라인을 count
```

```
Wend
Debug.Print cnt
Close #1
```

' (2) Rnd(); 1000 -> 800:200 나누기

While 1 ' While + 숫자; 조건이 만족될 때까지 계속 반복

Close #1, #2, #3 ' Output(특히), Input 파일은 항상 닫아줘야 함

cnt = 0 ' 초

cnttr = 0 ' 기

cntte = 0 ' 화

Open "C:\Users\USER\For 통특\Wex.txt" For Input As #1 ' 1000

Open "C:\Users\USER\For 통특\Wcnttr.txt" For Output As #2 ' 800

Open "C:\Users\USER\For 통특\Wcntte.txt" For Output As #3 ' 200

While Not EOF(1)

Input #1, a, b, c

cnt = cnt + 1 ' #1이 몇 줄인지 세기 위해서

Randomize

rd = Int(Rnd() * 10 + 1) ' 1-10 사이 난수 발생

Select Case rd

Case 1 To 8 ' 1-8 사이 값이면

Print #2, a, b, c ' #2 파일에 저장

cnttr = cnttr + 1 ' #2가 몇 줄인지 세기 위해서

Case 9 To 10

Print #3, a, b, c ' #3 파일에 저장

cntte = cntte + 1 ' #3이 몇 줄인지 세기 위해서

End Select

Wend

Debug.Print cnt, cnttr, cntte

If cnttr = 800 Then End ' 조건; 800이 되면 끝

Wend ' 조건(if문)이 만족될 때까지 계속 돌아감

End

End Sub

활용5-(3); 확률: 동전던지기

#1; 동전 앞면/뒷면이 나올 확률 by Rnd

Sub kk()

coin_num = 100 ' 동전던지기 횟수; 횟수가 많을수록 이론적 확률과 가까움

For i = 1 To coin_num

Randomize

coin = Int(Rnd * 2) ' 0(H) or 1(T)

Select Case coin

```

Case 0: H = H + 1 ' 앞면이 나온 횟수
Case 1: T = T + 1 ' 뒷면이 나온 횟수
End Select
Next i

Debug.Print "앞면이 나올 확률: "; H / coin_num
Debug.Print "뒷면이 나올 확률: "; T / coin_num

End Sub

```

11week

활용5-(4); 평균, 분산

- **평균**; $\mu = \frac{\sum x_i}{N}$ (ex. cm)
- **분산**; $\sigma^2 = \frac{\sum (x_i - \mu)^2}{N} = \frac{\sum x_i^2 - (\sum x_i)^2 / N}{N}$ (ex. cm²)
- **표준편차**; $\sigma = \sqrt{\text{분산}}$ (ex. cm)
- **공분산**; $Cov(X, Y) = \frac{\sum (x_i - \mu)(y_i - \mu)}{N}$ (단위 다름)
- **상관계수**; $Corr(X, Y) = \frac{Cov(X, Y)}{\sigma_x \sigma_y}$ (단위 없음; 공분산의 표준화)

#1; 평균

```

Sub kk()
Open "C:\Users\USER\For 통특\ex.txt" For Input As #1
While Not EOF(1)
Input #1, idd, itm, rat
cnt = cnt + 1 ' 총 몇 줄인지
sum_rat = sum_rat + rat ' rat 총합
Wend
Debug.Print sum_rat / cnt ' 평균
End
End Sub

```

#2; ID별 rate의 평균

```

Const idds = 10 ' 상수선언; 나중에 숫자가 바뀔 때 유용
Dim cnt_idd(idds)
Dim sum_idd(idds)
Dim mean_idd(idds)
Sub kk()
Open "C:\Users\USER\For 통특\ex.txt" For Input As #1
Open "C:\Users\USER\For 통특\ex_mean.txt" For Output As #2
While Not EOF(1)
Input #1, idd, itm, rat
cnt_idd(idd) = cnt_idd(idd) + 1 ' 각 id별 개수

```

```

    sum_idd(idd) = sum_idd(idd) + rat ' 각 id별 평점
Wend
For i = 1 To idds
    mean_idd(idd) = sum_idd(i) / cnt_idd(i)
    Print #2, i, mean_idd(idd) ' 각 id별 평균
Next i
End
End Sub

```

#3; 분산 by 편차의 평균

```

Sub kk()
    Open "C:\Users\WUSER\For 통특\Wex.txt" For Input As #1
    While Not EOF(1)
        Input #1, idd, itm, rat
        cnt = cnt + 1
        sum_rat = sum_rat + rat
    Wend
    mean_rat = sum_rat / cnt ' 평균
    Debug.Print mean_rat
Close #1
    Open "C:\Users\WUSER\For 통특\Wex.txt" For Input As #1
    While Not EOF(1)
        Input #1, idd, itm, rat
        sum_sq = sum_sq + (rat - mean_rat) ^ 2 ' 편차제곱합
    Wend
    var_rat = sum_sq / cnt ' 분산
    Debug.Print var_rat
End
End Sub

```

#4; 분산 by (제공의 평균) - (평균의 제공) * 추천

```

Sub kk()
    Open "C:\Users\WUSER\For 통특\Wex.txt" For Input As #1
    While Not EOF(1)
        Input #1, idd, itm, rat
        cnt = cnt + 1
        sum_rat = sum_rat + rat ' rate의 합
        sum_rat2 = sum_rat2 + (rat) ^ 2 ' rate의 제곱합
    Wend
    mean_rat = sum_rat / cnt
    var_rat = (sum_rat2 / cnt) - (mean_rat) ^ 2
    Debug.Print mean_rat, var_rat
End
End Sub

```

#5; ID별 rate의 분산

```

Const idds = 10 ' 상수 선언
Dim cnt(idds) ' 사실 11개의 방; 0번방은 비었다고 생각
Dim sum_rat(idds)
Dim sum_rat2(idds)
Sub kk()
    Open "C:\Users\WUSER\For 통특\Wex.txt" For Input As #1
    Open "C:\Users\WUSER\For 통특\Wex_var.txt" For Output As #2
    While Not EOF(1)
        Input #1, idd, itm, rat
        cnt(idd) = cnt(idd) + 1 ' 각 id의 개수
        sum_rat(idd) = sum_rat(idd) + rat ' 각 id의 rate 합
        sum_rat2(idd) = sum_rat2(idd) + (rat) ^ 2 ' 각 id의 rate 제곱합
    Wend
    For i = 1 To idds
        Print #2, i; cnt(i), sum_rat(i), sum_rat2(i) ' 추후 분석을 위해 저장해놓기
        Print #2, i; ","; (sum_rat2(i) - ((sum_rat(i)) ^ 2 / cnt(i))) / cnt(i) ' 각 id의 분산
    Next i
End
End Sub

```

활용5-(5); 소수인지 아닌지 찾기

' 소수; 1과 자신만으로 나뉘떨어지는 1보다 큰 정수
 ' 약수; 어떤 수를 나뉘떨어지게 하는 수
 ' Mod; 나머지 값

#1; 소수 찾기

```

Sub kk()
    ' idea; 나뉘떨어지는 수가 2개이면 소수
    ' InputBox(); 사용자 입력을 받는 함수
    ' Val(); 문자를 숫자로 변환
    x = Val(InputBox("숫자를 입력하십시오"))
    For i = 1 To x
        If (x Mod i) = 0 Then cnt = cnt + 1
    Next i
    If cnt = 2 Then
        Debug.Print x; "는 소수임"
    Else
        Debug.Print x; "는 소수 아님"
    End If
End
End Sub

```

#2; 약수 찾기

```

Sub kk()
    ' idea; 절반을 봤을 때 나뉘떨어지는 게 하나 이상 있으면 약수
    ' Math.Sqrt(); 루트

```



```

x = Val(InputBox("숫자를 입력하시오"))
For i = 1 To Int(Math.Sqr(x)) ' 효율성을 위해
    If (x Mod i) = 0 Then cnt = cnt + 1
Next i
If cnt = 1 Then
    Debug.Print x; "는 약수 아님"
Else
    Debug.Print x; "는 약수임"
End If
End Sub

```

Q. 앞의 절반을 보는 건데 왜 루트를 씌우는 건지?

#3; 가장 큰 소수 찾기

```

Sub kk()
' idea; 소수를 계속 찾다가 맨 마지막에 찾은 소수가 가장 큰 소수
' Math.Sqrt(); 루트
For i = 1 To 10000 ' 원하는 범위 내에서
    cnt = 0 ' cnt 초기화
    For j = 1 To i ' 각각의 숫자마다 소수인지 아닌지 확인
        If (i Mod j) = 0 Then cnt = cnt + 1
    Next j
    If cnt = 2 Then p_num = i ' 계속 갱신됨; 최종적으로 가장 큰 수가 저장
Next i
Debug.Print p_num
End
End Sub

```

활용6

```

' 알고리즘; 협업적 필터링
' Data set; grouplens.org
' u.data, u.item, u.user -> 3가지 데이터 셋을 이용해서 앞으로 분석할거임

```

1. 추천시스템: 엑셀 작업
2. 자료 변환
3. 자료 추출
4. 고객별 영화 수
5. 영화별 고객 수
6. 변수 추가
7. 분류
8. 전체 평가 점수의 평균
9. 고객별 평가 점수의 평균
10. 이웃 수
11. 두 고객 간의 상관계수