

別紙4ー46

「for」を使った反復構造の書き方の例

基本形	For 変数名 = 初期値 To 終値 Step 加算数 処理 Next	(Subプロシージャは省略)
例	Dim i As Integer For i = 1 To 5 Step 1 MsgBox i Next	iを1ずつ増やしながら1以上5以下の範囲で「MsgBox i」をくりかえす (1, 2, 3, 4, 5が表示される)
	Dim i As Integer For i = 3 To 5 Step 1 MsgBox i Next	iを1ずつ増やしながら3以上5以下の範囲で「MsgBox i」をくりかえす (3, 4, 5が表示される)

別紙4ー47

TRY「プログラムの作成5（二分法探索）」のデータ（Python）

```
data = [2, 5, 8, 9, 30, 45, 55, 102, 183, 209]
value = 183
i = 0
left = 0
right = len(data) - 1
while left <= right:
    mid = (left+right+1) // 2
    if data[mid] == value:
        print(value, "を", mid+1, "番目に発見")
        print("くりかえし回数は", i+1)
        break
    elif data[mid] < value:
        left = mid
    else:
        right = mid - 1
    i = i + 1
```

別紙4ー48

TRY「プログラムの作成5（二分法探索）」のデータ（表計算マクロ）

```
Sub 二分法探索()
Dim Data As Variant
Dim value As Integer
Dim i As Integer
Dim left As Integer
Dim right As Integer
Dim mid As Integer

Data = Array(2, 5, 8, 9, 30, 45, 55, 102, 183, 209)
value = 183
i = 0
left = 0
right = UBound(Data)

Do While left <= right
    mid = (left + right + 1) \ 2
    If Data(mid) = value Then
        MsgBox value & "を" & (mid + 1) & "番目に発見"
        MsgBox "くりかえし回数は" & (i + 1)
        Exit Do
    ElseIf Data(mid) < value Then
        left = mid
    Else
        right = mid - 1
    End If
    i = i + 1
Loop
End Sub
```

別紙4ー49

TRY「プログラムの作成5（二分法探索）」のデータ（疑似言語）

```
data = [2, 5, 8, 9, 30, 45, 55, 102, 183, 209]
value = 183
i = 0
left = 0
right = 長さ(Data) - 1
while left <= right の間くりかえす:
    mid = (left+right+1) ÷ 2
    もし Data[mid] = value ならば:
        表示する (value, "を", mid+1, "番目に発見")
        表示する ("くりかえし回数は", i+1)
        (くりかえしをやめる処理)
    そうでなくもし Data[mid] < value ならば:
        left = mid
    そうでなければ:
        right = mid - 1
    i = i + 1
```

※ここでは、下記の「長さ」という関数を作成し使用する。
長さ（配列名）…引数として与えられた配列名の配列の要素数を数える関数

別紙4ー50

二分法探索のプログラムの解説

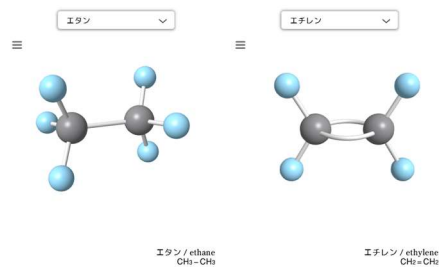
探索する値 (value)	添字	0	1	2	3	4	5	6	7	8	9
183	配列data	2	5	8	9	30	45	55	102	183	209
											len(data)-1

回数	i	left	right	mid	data[mid]	実行される処理
1回目	0	0	9	5	45	13行目の処理 left=mid
2回目	1	5	9	7	102	13行目の処理 left=mid
3回目	2	7	9	8	183	9～11行目の処理 (終了)

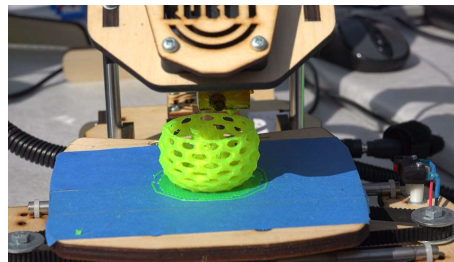
別紙4ー51



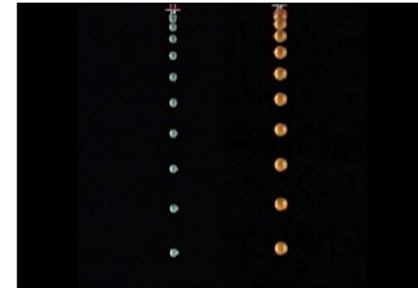
別紙4ー52



別紙4ー53

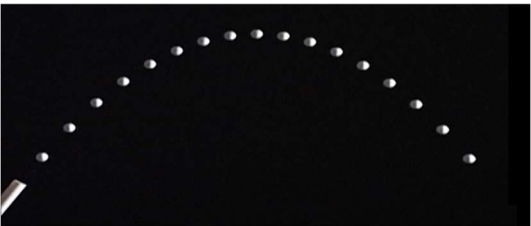


別紙4ー54



[illegible]

別紙4-56

[illegible]

print関数の使い方

基本形	print(引数) : 引数の数値や文字列を画面に出力する
例	意味
print(10)	画面に10を出力する
print("こんにちは")	画面に「こんにちは」という文字を出力する
print(10,"回")	画面に「10」という文字を出力する (カンマ ,) で区切った複数の文字を指定できる
i = 5 print(i,"回")	変数iに5を代入し、画面に「5 回」という文字を出力する

```
TRY「ボールの投げ上げのシミュレーション」のデータ (Python)
```

```
x = 0
y = 0
vx = 7
vy = 5
t = 0
while y >= 0:
    print(t, "(", x, ", ", y, ")")
    x = x + vx
    y = y + vy
    vy = vy - 1
    t = t + 1
```

```
TRY「ボールの投げ上げのシミュレーション」のデータ（疑似言語）

x = 0
y = 0
vx = 7
vy = 5
t = 0
y >= 0 の間くりかえす：
    表示する(t, "(", x, ", ", y, ")")
    x = x + vx
    y = y + vy
    vy = vy - 1
L t = t + 1
```

```
import random
i = 1
score = 0
while i <= 3:
    if random.random() < 0.5:
        print("ハート")
        score = score + 1
    else:
        print("スベード")
        i = i + 1
print(score, "点")
```

```

TRY「トランプを使った占い」のデータ（疑似言語）

i = 1
score = 0
i <= 3 の間くりかえす：
  もし 乱数() < 0.5 ならば：
    表示する（“ハート”）
    score = score + 1
  そうでなければ：
    表示する（“スベード”）
  i = i + 1
表示する（score, “点”）

```

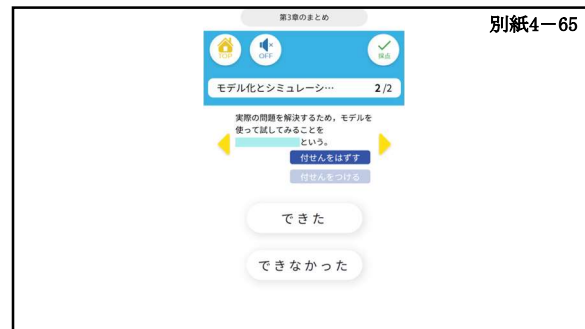
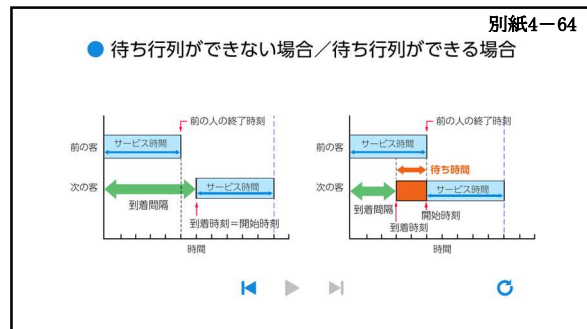
※ここでは、下記の「乱数」という関数を作成し使用する。
 乱数 () …0以上1未満のランダムな数を出力する関数

複合代入演算子の種類と使用例

演算子	意味	使用例	使用例の意味
+=	たし算	a+=3	変数aに3をたす
-=	ひき算	a-=5	変数aから5をひく
=	かけ算	a=11	変数aに11をかける
/=	わり算	a/=3	変数aを3でわる

- ・ 複合代入演算子は、累算代入演算子ともよばれる。
- ・ 代入演算子と算術演算子を使った記述を短くしたいときに使用される。
たとえば、`a+=3` は `a=a+3` と同じ意味である。

別紙4-63



別紙4ー66

実習8「Pythonでアルゴリズムを学ぼう」のデータ (Python)

```

motonosu = 2
kuwaerusu = 0.00001
heihokon1 = 0
heihokon2 = 0

while heihokon2**2 < motonosu:
    heihokon1 = heihokon2
    heihokon2 += kuwaerusu

print(heihokon1)
print(heihokon2)

```

#平方根の近似値を求めたい値を2として代入する
#加える数を0.00001として代入する
#平方根の近似値を、初期値を0として代入する
#2乗した数が求めたい数より小さいときくりかえし
#くりかえして小さい数をたす前の数を変数に代入する
#小さな数をたす
#近似値が含まれる範囲の下限の数を表示
#近似値が含まれる範囲の上限の数を表示

別紙4ー67

実習8「Pythonでアルゴリズムを学ぼう」のデータ (疑似言語)

```

motonosu = 2
kuwaerusu = 0.00001
heihokon1 = 0
heihokon2 = 0

heihokon2**2 < motonosu の間くりかえす:
  heihokon1 = heihokon2
  heihokon2 += kuwaerusu

表示する(heihokon1)
表示する(heihokon2)

```

別紙4ー68

実習9「待ち行列のシミュレーション」のデータ (Python)

```

import random
nintzu = 10

r_kankaku = 0
r_service = 0
p_toclime = 0
p_end = 0
p_start = 0
s_waitsum = 0
i = 0

while i <= nintzu-1:
    print("客番号:", i, "...")
    r_kankaku = int(random.random()*10)
    print("客の到着間隔:", r_kankaku)
    p_toclime = 0
    r_service = 1 + int(random.random()*5)
    print("サービス時間:", r_service)
    if p_toclime > p_end:
        p_start = p_toclime
    else:
        p_start = p_end
    print("サービス開始時刻:", p_start)
    p_end = p_start + r_service
    print("サービス終了時刻:", p_end)

```

#Randomモジュールのインポート
#到着する客の人数
#次の客の到着間隔を、初期値を0として代入する
#サービス時間を、初期値を0として代入する
#客ごとの到着時刻を、初期値を0として代入する
#前の客のサービス終了時刻を、初期値を0として代入する
#客ごとのサービス開始時刻を、初期値を0として代入する
#待ち時間総和を、初期値を0として代入する
#くりかえしの範囲を決める変数の初期値を0として代入する
#人数分くりかえし
#客ごとに区切りを表示
#次の客の到着間隔を乱数で決める (0~9分とする)
#次の客の到着時刻を表示
#到着時刻を客に
#サービス時間を乱数で決める (1~5分とする)
#サービス時間を表示
#もし到着時刻が前の客のサービス終了時刻より遅ければ
#サービス開始時刻は到着時刻と同じ (待たない)
#そうでなければ、(到着時刻の遅れ分)
#サービス開始時刻は前の客のサービス終了時刻になる
#サービス開始時刻を表示
#この客のサービス終了時刻を求める
#サービス終了時刻を表示

別紙4ー69

実習9「待ち行列のシミュレーション」のデータ (疑似言語)

```

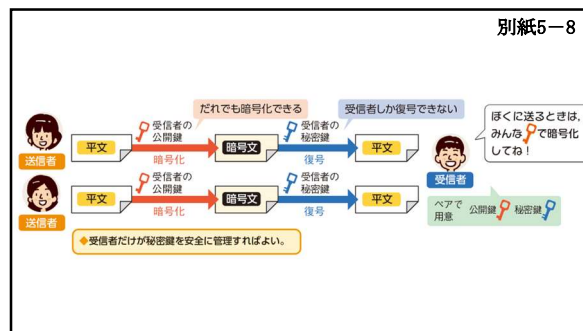
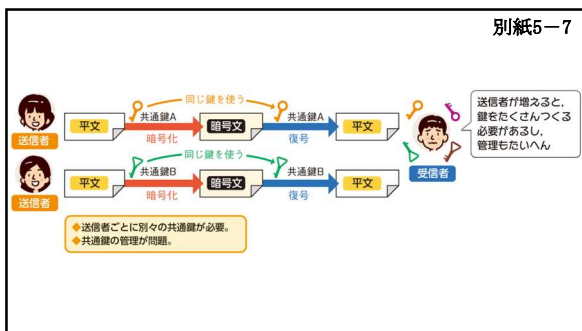
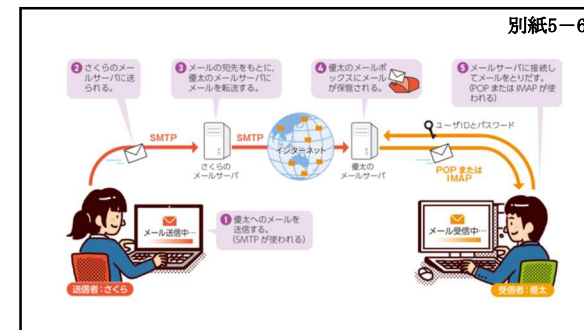
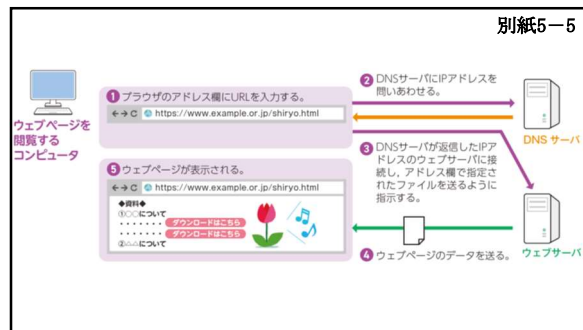
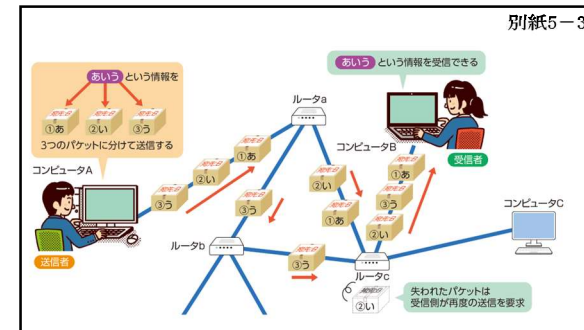
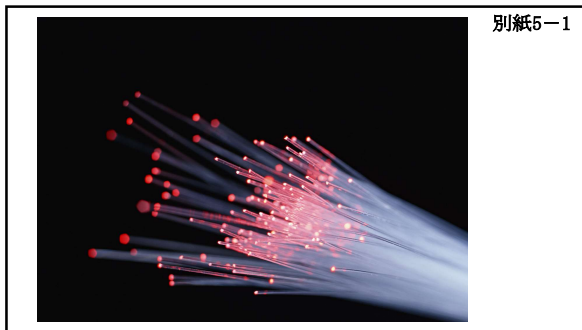
nintzu = 10

r_kankaku = 0
r_service = 0
p_toclime = 0
p_end = 0
p_start = 0
s_waitsum = 0
i = 0

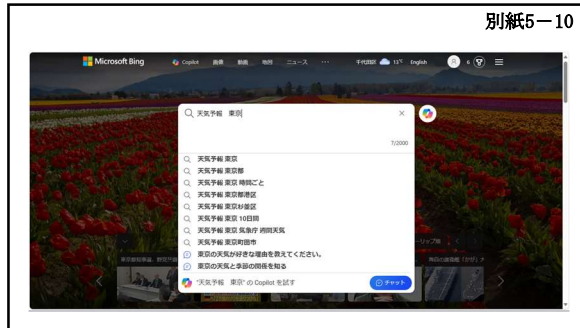
i <= nintzu-1 の間くりかえす:
  表示する("客番号:", i, "...")
  r_kankaku = 整数(乱数()*10)
  表示する("次の客の到着間隔:", r_kankaku)
  表示する("到着時刻:", p_toclime)
  r_service = 1 + 整数(乱数()*5)
  表示する("サービス時間:", r_service)
  もし p_toclime > p_end ならば:
    p_start = p_toclime
  そうでなければ:
    p_start = p_end
  表示する("サービス開始時刻:", p_start)
  p_end = p_start + r_service
  表示する("サービス終了時刻:", p_end)

```

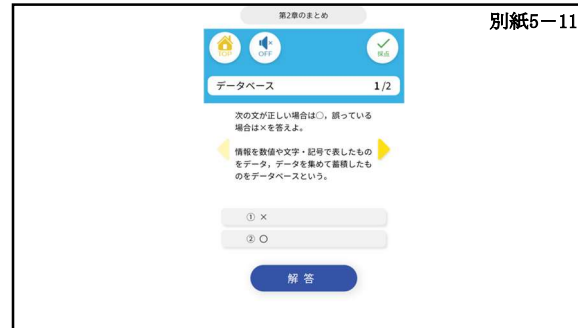




別紙5-10



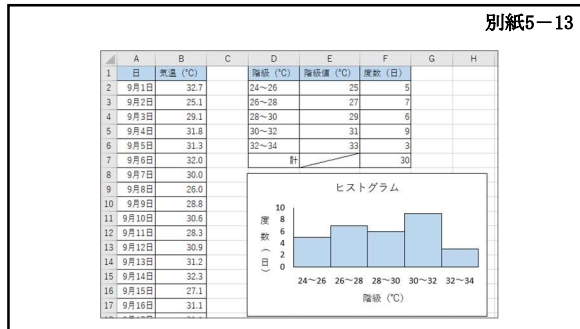
別紙5-11



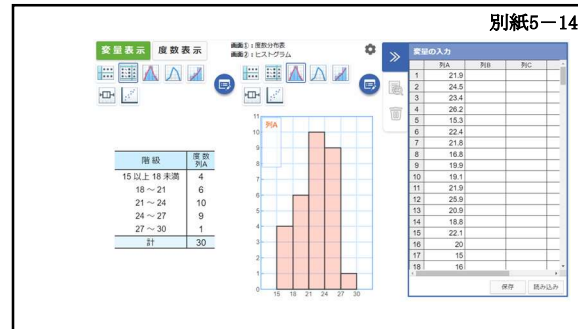
別紙5-12

日	気温 (°C)
9月1日	32.7
9月2日	25.1
9月3日	29.1
9月4日	31.8
9月5日	31.3
9月6日	32.0
9月7日	30.0
9月8日	28.0
9月9日	28.8
9月10日	30.9
9月11日	28.3
9月12日	30.9
9月13日	31.2
9月14日	32.3
9月15日	27.1
9月16日	31.1
9月17日	27.1
9月18日	26.7
9月19日	26.0
9月20日	26.0
9月21日	26.0
9月22日	26.0
9月23日	26.0
9月24日	26.0
9月25日	26.0
9月26日	26.0
9月27日	26.0
9月28日	26.0
9月29日	26.0
9月30日	26.0

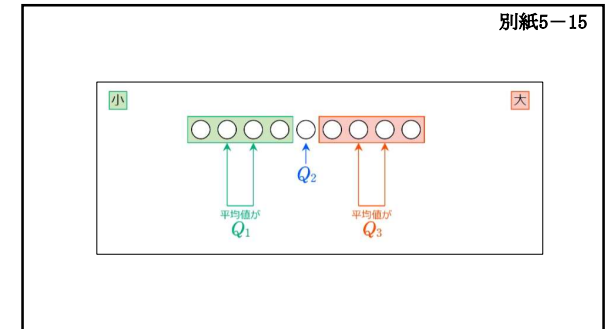
別紙5-13



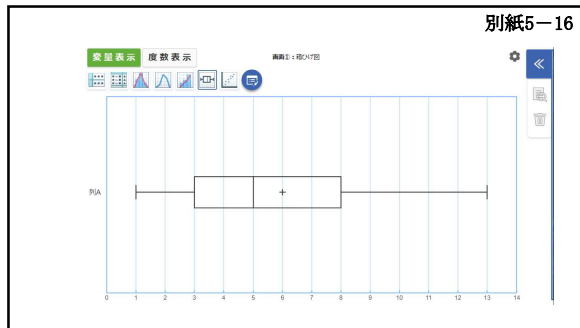
別紙5-14



別紙5-15



別紙5-16



別紙5-17

元データ	年代	評価
中学生	2	
高校生	3	
大学生	2	
大学生	3	
高校生	3	
大学生	3	
中学生	1	
中学生	1	
高校生	2	
高校生	3	

別紙5-18

元データ	年代	評価
中学生	2	
高校生	3	
大学生	2	
高校生	3	
大学生	3	
中学生	1	
中学生	1	
高校生	2	
高校生	3	

別紙5-19

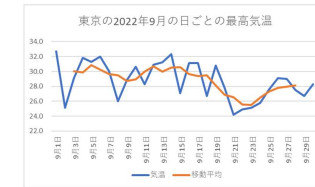
A町における商品の価格		B町における商品の価格	
店舗	価格 (円)	店舗	価格 (円)
店舗1	260	店舗1	100
店舗2	270	店舗2	260
店舗3	280	店舗3	270
店舗4	280	店舗4	280
店舗5	300	店舗5	280
		店舗6	280

別紙5-20

	A	B	C	D	E
1	A町における商品の価格		B町における商品の価格		
2	店舗	価格 (円)		店舗	価格 (円)
3	店舗1	260		店舗1	100
4	店舗2	270		店舗2	260
5	店舗3	280		店舗3	270
6	店舗4	280		店舗4	280
7	店舗5	300		店舗5	280
8				店舗6	280
9					
10	平均値	278		平均値	245
11	中央値	280		中央値	275

別紙5-21

移動平均の例



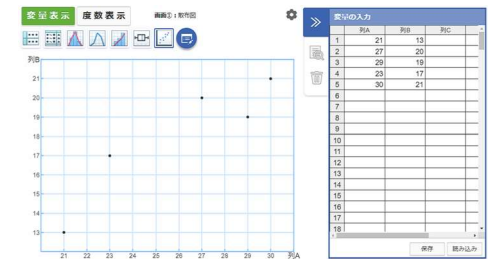
別紙5-22

10人の漢字テストの得点 (A組)			10人の漢字テストの得点 (B組)		
生徒	得点 (点)	(得点-平均値) ²	生徒	得点 (点)	(得点-平均値) ²
生徒1	9	4	生徒1	6	1
生徒2	3	16	生徒2	10	9
生徒3	4	9	生徒3	7	0
生徒4	10	9	生徒4	7	0
生徒5	10	9	生徒5	5	4
生徒6	5	4	生徒6	4	9
生徒7	7	0	生徒7	9	4
生徒8	9	4	生徒8	10	9
生徒9	10	9	生徒9	5	4
生徒10	3	16	生徒10	7	0
合計	70	80	合計	70	40

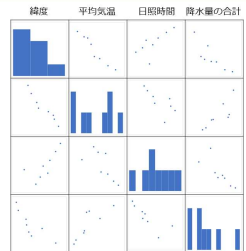
別紙5-23

	A	B	C	D	E	F	G
1	10人の漢字テストの得点 (A組)			10人の漢字テストの得点 (B組)			
2	生徒	得点 (点)	(得点-平均値) ²	生徒	得点 (点)	(得点-平均値) ²	
3	生徒1	9	4	生徒1	6	1	
4	生徒2	3	16	生徒2	10	9	
5	生徒3	4	9	生徒3	7	0	
6	生徒4	10	9	生徒4	7	0	
7	生徒5	10	9	生徒5	5	4	
8	生徒6	5	4	生徒6	4	9	
9	生徒7	7	0	生徒7	9	4	
10	生徒8	9	4	生徒8	10	9	
11	生徒9	10	9	生徒9	5	4	
12	生徒10	3	16	生徒10	7	0	
13	合計	70	80	合計	70	40	
14							
15	平均値	7		平均値	7		
16	分散	8		分散	4		
17	標準偏差	2.8		標準偏差	2.0		

別紙5-24



別紙5-25

散布図
行列の
例

別紙5-26

3. 分散 (ばらつき) の求め方

分散は、データのばらつきを数値で表す。分散は、平均値からの偏差の二乗の平均値である。

$$\text{分散} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

ここで、 x_i はデータの値、 $\bar{x}$ は平均値、 n はデータの総数である。

例として、次のデータの分散を計算する。

データ	1	2	3	4	5
値	1	2	3	4	5

平均値 $\bar{x} = 3$ である。

$$\text{分散} = \frac{1}{5} [(1-3)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2 + (5-3)^2] = \frac{1}{5} [4 + 1 + 0 + 1 + 4] = \frac{10}{5} = 2$$

別紙5-27

3. 分散 (ばらつき) の求め方

分散は、データのばらつきを数値で表す。分散は、平均値からの偏差の二乗の平均値である。

$$\text{分散} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

ここで、 x_i はデータの値、 $\bar{x}$ は平均値、 n はデータの総数である。

例として、次のデータの分散を計算する。

データ	1	2	3	4	5
値	1	2	3	4	5

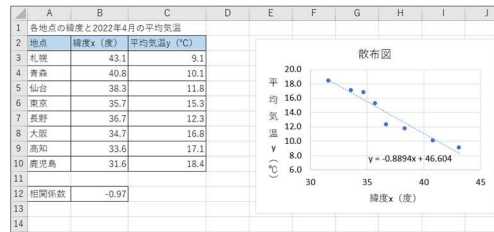
平均値 $\bar{x} = 3$ である。

$$\text{分散} = \frac{1}{5} [(1-3)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2 + (5-3)^2] = \frac{1}{5} [4 + 1 + 0 + 1 + 4] = \frac{10}{5} = 2$$

別紙5-28

	A	B	C	D	E	F	G	H	I
1	地点	札幌	青森	仙台	東京	長野	大阪	高知	鹿児島
2	緯度x (度)	43.1	40.8	38.3	35.7	36.7	34.7	33.6	31.6
3	平均気温y (°C)	9.1	10.1	11.8	15.3	12.3	16.8	17.1	18.4

別紙5-29



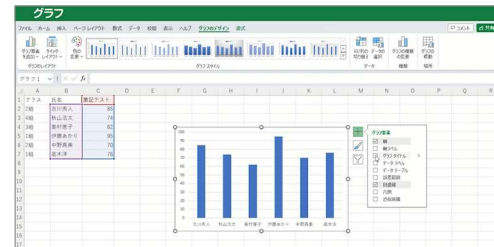
別紙5-30



別紙5-31

	日	月	火	水	木	金	土	合計	平均	売上枚の内訳
ブレンド	46	21	43	42	38	54	60	304	43	20%
アメリカン	51	27	40	48	42	46	55	309	44	20%
エスプレッソ	31	16	31	35	30	39	41	223	32	15%
カフェラテ	42	25	37	44	46	46	51	291	42	19%
カフェモカ	65	35	52	51	56	58	75	392	56	26%
合計	235	124	203	220	212	243	282	1,519	217	100%

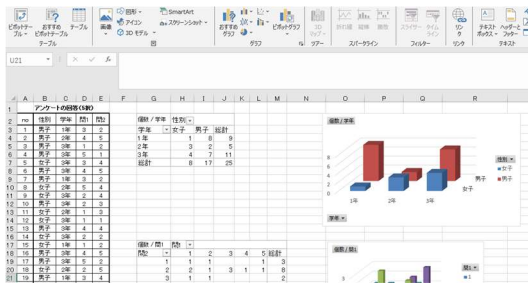
別紙5-32



別紙5-33

順位	性別	学年	国1	国2
1	男子	1年	3	2
2	男子	2年	4	3
3	男子	3年	1	2
4	男子	3年	3	1
5	女子	3年	3	4
6	男子	3年	4	3
7	男子	1年	3	2
8	女子	2年	5	4
9	女子	3年	2	4
10	男子	3年	2	3
11	女子	2年	1	3
12	女子	2年	1	1
13	男子	3年	4	4
14	女子	3年	2	2
15	女子	1年	1	2
16	男子	3年	4	3
17	男子	3年	5	2
18	女子	2年	2	3
19	男子	1年	3	4
20	男子	1年	4	2
21	男子	1年	1	4
22	男子	1年	2	1
23	男子	1年	4	4
24	男子	1年	1	3
25	男子	2年	3	2

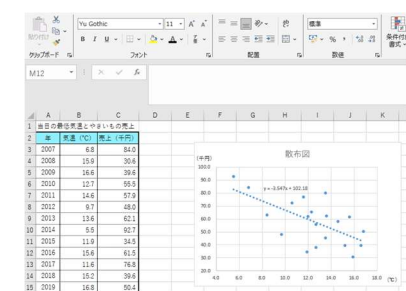
別紙5-34



別紙5-35

年	気温 (°C)	売上 (千円)
2007	6.8	84.0
2008	15.9	30.6
2009	16.6	39.6
2010	12.7	55.5
2011	14.6	57.9
2012	9.7	48.0
2013	13.6	62.1
2014	5.5	92.7
2015	11.9	34.5
2016	15.6	61.5
2017	11.6	76.8
2018	15.2	39.6
2019	16.8	50.4

別紙5-36



HOME

VOICE

OK

ネットワークのしくみ

1/18

用語：

説明：

さまざまな情報機器をつないだ通信システム。

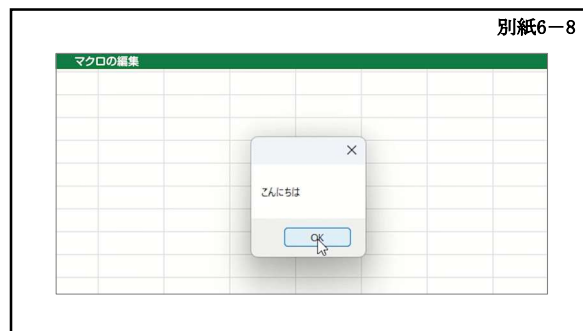
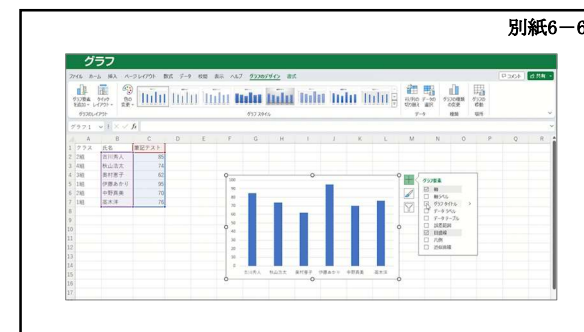
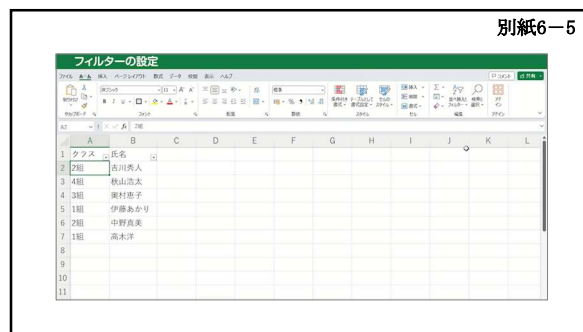
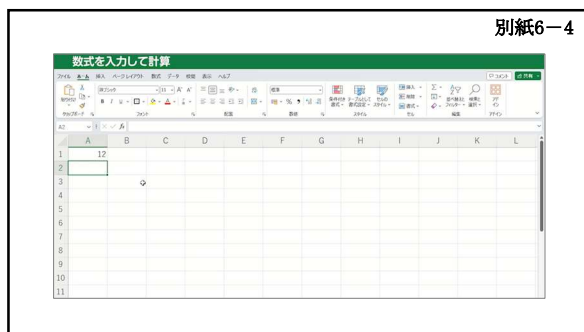
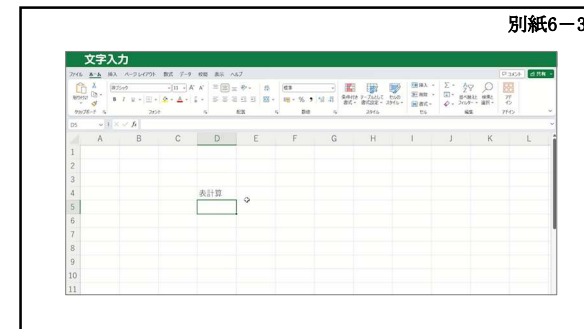
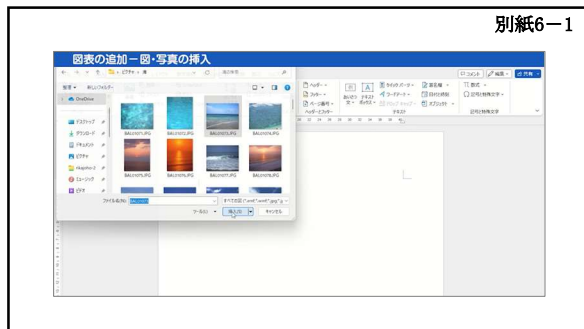
付せんをはすす

付せんをつける

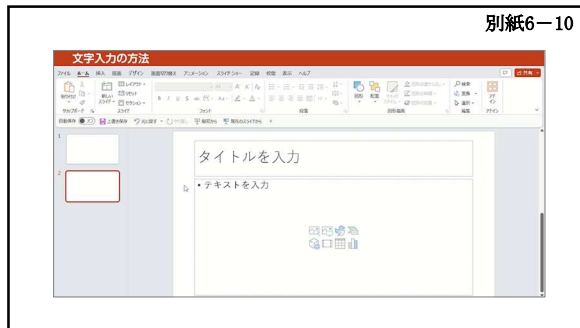
できた

できなかった

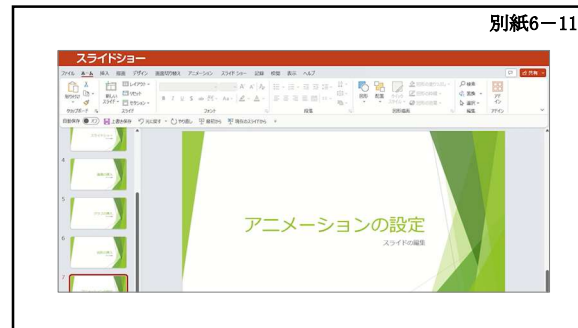
別紙5-37



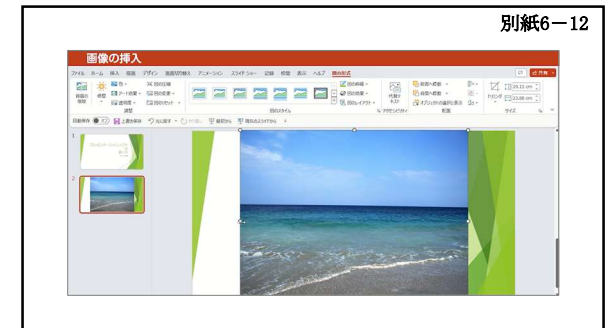
別紙6-10



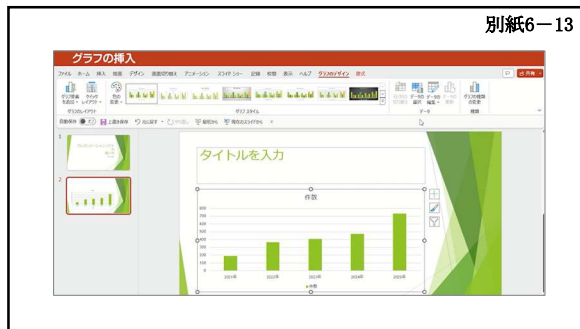
別紙6-11



別紙6-12



別紙6-13



別紙6-14



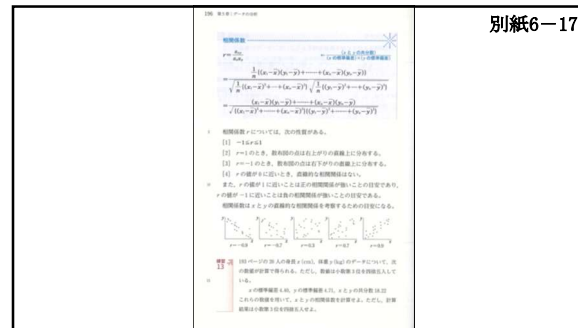
別紙6-15



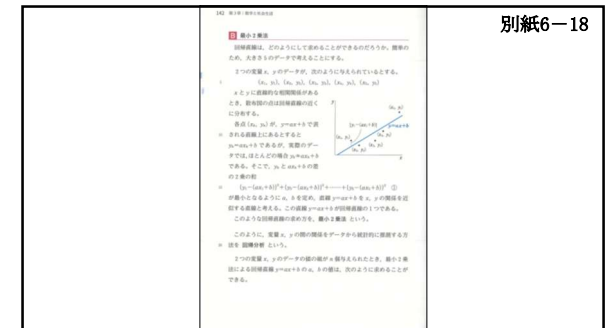
別紙6-16



別紙6-17



別紙6-18



別紙6－19

「分岐構造」のプログラムのデータ (Python)

```
n = 70
if n >= 80:
    print("最高")
elif n >= 60:
    print("よい")
else:
    print("普通")
```

別紙6－20

「分岐構造」のプログラムのデータ (疑似言語)

```
n = 70
もし n >= 80 ならば:
    | 表示する("最高")
elif n >= 60 ならば:
    | 表示する("よい")
else:
    | 表示する("普通")
```

別紙6－21

「反復構造」のプログラムのデータ (Python)

```
for i in range(1, 6, 2):
    print(i)

-----

for i in range(3, 6, 1):
    print(i)

-----

for i in range(3):
    print(i)

-----

for i in range(3):
    print("A")
```

別紙6－22

「反復構造」のプログラムのデータ (疑似言語)

```
iを1から5まで2ずつ増やしながらくりかえす:
| 表示する (i)

-----

iを3から5まで1ずつ増やしながらくりかえす:
| 表示する (i)

-----

iを0から2まで1ずつ増やしながらくりかえす:
| 表示する (i)

-----

iを0から2まで1ずつ増やしながらくりかえす:
| 表示する ("A")
```