

工業情報数理

1 学期期末

担当者：松井（自称 昨日まで20）

2

else if による多岐選択

例題

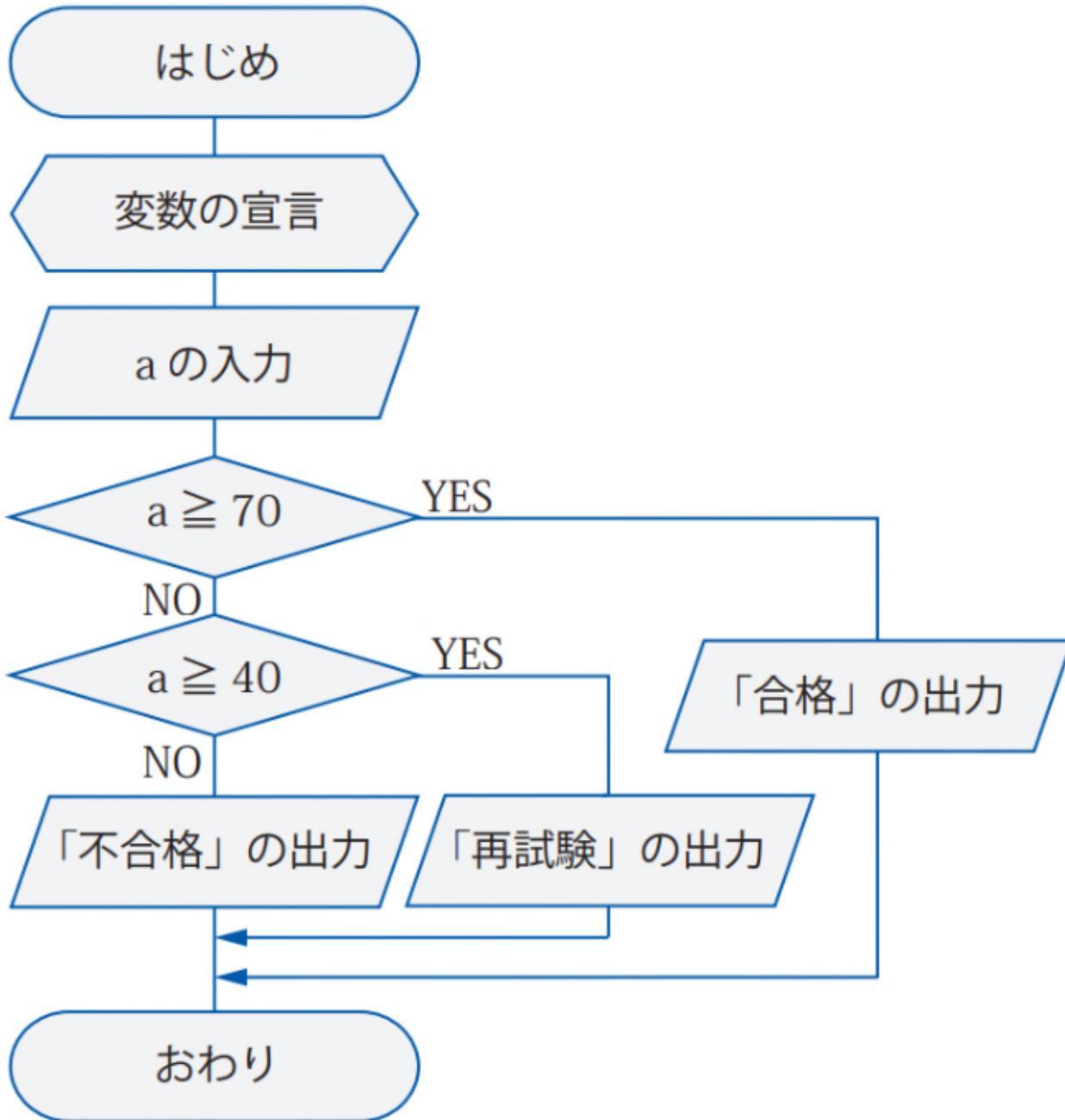
6

キーボードから入力したテストの点数が、70 点以上なら「合格」、40 点以上 69 点以下なら「再試験」、39 点以下なら「不合格」と出力するプログラムをつくりなさい。

45 
再試験

▲出力結果

```
01  /* 例題 6 */
02  #include<stdio.h>
03  int main(void)
04  {
05      int a;
06      scanf("%d",&a);
07      if(a >= 70) {
08          printf("合格\n");
09      }
10      else if(a >= 40) {
11          printf("再試験\n");
12      }
13      else {
14          printf("不合格\n");
15      }
16      return 0;
17  }
```



```
if (条件 1) {  
    条件 1 が真の場合に実行する文 1  
    条件 1 が真の場合に実行する文 2  
    ⋮  
}  
else if (条件 2) {  
    条件 1 が偽で、条件 2 が真の場合に実行する文 1  
    条件 1 が偽で、条件 2 が真の場合に実行する文 2  
    ⋮  
}  
else {  
    条件 1, 2 とも偽の場合に実行する文 1  
    条件 1, 2 とも偽の場合に実行する文 2  
    ⋮  
}
```

- あるデパートでは、買い上げ金額が 3000 円以上なら 1割引、5000 円以上なら 2 割引で販売している。キーボードから買い上げ金額を入力して、支払い金額を計算するプログラムをつくりなさい。ただし、消費税は考えないものとする。

```
#include <stdio.h>
int main(void){
    float a;
    printf(" 買い上げ金額の入力 ¥n");
    scanf("%f",&a);
    if(a >= 5000){
        printf(" 2 割引  %6.0f¥n",a*0.8);
    }
    else if(a >= 3000){
        printf(" 1 割引  %f¥n",a*0.9);
    }
    else{
        printf("割引なし  %f¥n",a);
    }
    return 0;
}
```

繰返し処理

同じ処理を複数回繰り返す場合には、繰返し処理を行う制御文を用いると便利である。この節では、for 文と while 文、do ~ while 文を学ぶ。

1 for 文

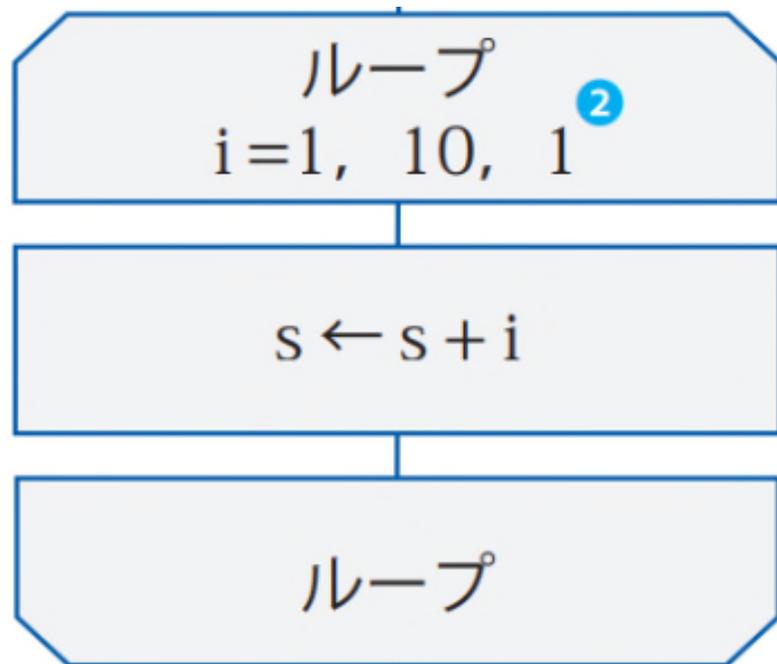
for 文は、初期値、条件、増分を指定して繰返し^①処理を行う制御文である。

例題

8

1 から 10 までの整数の和を計算するプログラムをつくりなさい。

s = 55



```
int i, s;
s = 0;
for(i = 1; i <= 10; i++) {
    s = s + i;
}
printf("s = %d\n", s);
return 0;
```

```
for (初期値 ; 条件 ; 増分) {  
    実行文 1  
    実行文 2  
    ⋮  
}
```

```
int i, s;  
s = 0;  
for(i = 1; i <= 10; i++) {  
    s = s + i;  
}  
printf("s = %d¥n", s);  
return 0;
```

```
#include <stdio.h>
int main(void){
    int i,s;
    s=0;
    for (i = 1; i<=10 ; i++){
        s=s+i;
        printf("i=%d,s=%d\n",i,s);
    }
    printf("\ni=%d,s=%d\n\n",i,s);
return 0;}
```

```
i=1, s=1
i=2, s=3
i=3, s=6
i=4, s=10
i=5, s=15
i=6, s=21
i=7, s=28
i=8, s=36
i=9, s=45
i=10, s=55

i=11, s=55
```

```
#include <stdio.h>
int main(void){
    int i,s;
    s=0;
    for (i = 1; i<10 ; i++){
        s=s+i;
        printf("i=%d,s=%d\n",i,s);
    }
    printf("\ni=%d,s=%d\n\n",i,s);
    return 0;
}
```

```
i=1, s=1
i=2, s=3
i=3, s=6
i=4, s=10
i=5, s=15
i=6, s=21
i=7, s=28
i=8, s=36
i=9, s=45

i=10, s=45
```

```
#include <stdio.h>
int main(void){
    int i,s;
    s=0;
    for (i = 1; i<=10 ; i=i+2){
        s=s+i;
        printf("i=%d,s=%d\n",i,s);
    }
    printf("\ni=%d,s=%d\n\n",i,s);
    return 0;
}
```

```
i=1, s=1
i=3, s=4
i=5, s=9
i=7, s=16
i=9, s=25

i=11, s=25
```

for 文に関する問題

初期値：1 最終値：10 増加分：1
for (i = ; i <= ; i = i +)

初期値：1 最終値：10 増加分：2
for (i = ; i <= ; i = i +)

初期値：2 最終値：10 増加分：2
for (i = ; i <= ; i = i +)

初期値：1 最終値：10 增加分：1
for (i = 1 ; i <= 10 ; i = i + 1)

初期値：1 最終値：10 增加分：2
for (i = 1 ; i <= 10 ; i = i + 2)

初期値：2 最終値：10 增加分：2
for (i = 2 ; i <= 10 ; i = i + 2)

for 文に関する問題

初期値：1 最終値：10 増加分：1
for (i = ; i < ; i = i +)

初期値：1 最終値：10 増加分：2
for (i = ; i < ; i = i +)

初期値：10 最終値：1 増加分：1
for (i = ; i >= ; i = i -)

初期値 : 1 最終値 : 1 0 增加分 : 1
for (i = 1 ; i < 1 1 ; i = i + 1)

初期値 : 1 最終値 : 1 0 增加分 : 2
for (i = 1 ; i < 1 1 ; i = i + 2)

初期値 : 1 0 最終値 : 1 增加分 : 1
for (i = 1 0 ; i > = 1 ; i = i - 1)

```
#include <stdio.h>
int main(void){
    int i,s;
    s=0;
    for (i = 10; i>=1 ; i=i-1){
        s=s+i;
        printf("i=%d,s=%d\n",i,s);
    }
    printf("\ni=%d,s=%d\n\n",i,s);
    return 0;
}
```

```
i=10, s=10
i=9, s=19
i=8, s=27
i=7, s=34
i=6, s=40
i=5, s=45
i=4, s=49
i=3, s=52
i=2, s=54
i=1, s=55

i=0, s=55
```

1 から 10 までの奇数の和を求める。

初期値：1 最終値：10 増加分：2
for (i = ; i <= ; i = i +)

2 から 10 までの偶数の和を求める。

初期値：2 最終値：10 増加分：2
for (i = ; i <= ; i = i +)

2

while 文

繰り返す回数があらかじめわかっていないときでも使用できる繰り返し制御文として **while 文**がある。

例題

9

キーボードから入力した整数の和を計算し、入力が終了したときに最終合計を出力するプログラムをつくりなさい。ただし 999 を入力したときに、データの入力を終了するものとし、999 は計算結果には加算しない。

```
12 ↵  
7 ↵  
24 ↵  
999 ↵  
s = 43
```

▲出力結果

考え方

while 文の繰返し条件を、「入力データ ≠ 999」とし、999 が入力されるまで合計を計算する。

```
#include <stdio.h>
int main(void){
    int i,s;
    scanf("%d",&i);
    s=0;
    while(i != 999){
        s=s+i;
        scanf("%d",&i);
    }
    printf("s=%d\n",s);
    return 0;
}
```

実行結果

```
1  
2  
999  
s=3
```

A**while 文**

while 文は，繰り返す条件を指定して使用する繰り返し制御文であり，次の書式で記述する。

```
while (条件) {  
    実行文 1  
    実行文 2  
    ⋮  
}
```

[例] `while(i<100) {
 i++;
}`

B**無限ループ****tips**

while 文の条件を 1 にすると、いつまでも繰り返し処理を続ける無限ループ^①を構成できる。while 文を用いた無限ループは、次の書式で記述する。

```
while (1) {  
    実行文 1  
    実行文 2  
    ⋮  
}
```

【例】 while(1) {
 実行文 1
 if(条件)break;
 実行文 2
 ⋮
}

**Tips**

Cでは、0を偽(条件不成立)、0以外を真(条件成立)と定義している。

① infinite loop
永久ループともいう。

while 文を終了し、無限ループから抜けるには、break 文を実行する。上の例では、if 文の条件が真の場合に、break 文が実行され、while 文による無限ループを終了する。

1 から 10 までの整数の和を求めるプログラムを while 文を使用してつくりなさい。

```
#include <stdio.h>
int main(void){
    int i,s;
    s=0;
    while(i <= 10){
        s=s+i;
        i++;
    }
    printf("s=%d¥n",s);
return 0;
}
```

while 文による無限ループ

```
#include <stdio.h>
int main(void){
    int i,s;
    i=1;
    s=0;
    while(1){
        s=s+i;
        i++;
    }
    printf("s=%d¥n",s);
    printf("i=%d¥n",i);
    return 0;
}
```

```
#include <stdio.h>
int main(void){
    int i,s;
    i=1;
    s=0;
    while(    ){
        s=s+i;
        i++;
        if(    )    ;
        printf("s=%d¥n",s);
        printf("i=%d¥n",i);
    }
    printf("s=%d¥n",s);
    printf("i=%d¥n",i);
    return 0;
}
```

3

do ~ while 文

実行文を処理したあとに繰返し条件を判断する制御文として **do ~ while 文**がある。

例題

10

キーボードから入力した整数を 1 行ずつ画面に出力するプログラムをつくりなさい。ただし 999 を入力したときに、データの入力を終了するものとし、画面に「 $i = 999$ 」と出力する。

考え方

do ~ while 文の繰返し条件を、「入力データ $\neq 999$ 」とし、999 が入力されるまで整数を画面に表示する。999 が入力された場合には、画面に $i = 999$ と表示して do ~ while 文を終了する。このため、出力処理は少なくとも 1 回実行されることになる。

```
55 ↵  
i = 55  
74 ↵  
i = 74  
38 ↵  
i = 38  
999 ↵  
i = 999
```

▲出力結果

ループ

i の入力

i の出力

ループ

i = 999

```
01  /* 例題 10 */
02  #include<stdio.h>
03  int main(void)
04  {
05      int i;
06      do {
07          scanf("%d",&i);
08          printf("i = %d¥n",i);
09      }while(i != 999);
10      return 0;
11  }
```

```
#include <stdio.h>
int main(void){
    int i;
    do{
        scanf("%d",&i);
        printf("i=%d¥n",i);
    }while(i != 999);
    return 0;
}
```

1

i=1

2

i=2

3

i=3

999

i=999

do ~ while 文では、はじめから繰り返し条件が成立していない場合であっても、実行文は最低一回実行される。このため、少なくとも一回は実行文を実行させたいときに、do ~ while 文を用いる。do ~ while 文は、次の書式で記述する。

```
do {  
    実行文 1  
    実行文 2  
    ⋮  
} while (条件);
```

[例] do {
 i++;
 } while(i<100);

練習問題

10

図6に示す二つのプログラムの実行結果を比較しなさい。

```
01  /* 練習問題 10-1 */
02  #include<stdio.h>
03  int main(void)
04  {
05      int i;
06      i = 10;
07      while(i < 5){
08          printf("ELIZABETH¥n");
09      }
10      return 0;
11 }
```

```
01  /* 練習問題 10-2 */
02  #include<stdio.h>
03  int main(void)
04  {
05      int i;
06      i = 10;
07      do{
08          printf("ELIZABETH¥n");
09      }while(i < 5);
10      return 0;
11 }
```