

計算機プログラミング

第6回目

May 20, 2021

(10:30～12:00)



担当教員: Thi Thi Zin (ティティズイン)
<thithi@cc.miyazaki-u.ac.jp>



本日の内容

- 前回の内容について復習
- for 文について
- 繰り返し処理 -while文-
- 課題 (レポート提出あり)

提出締切: 5月22日(土)20:00 まで

出席条件

- 毎回、締め切りまでに問題の解答をレポートとして提出する
- 手書きをPDFファイルで提出することも可能
- レポートには日付、自分の名前、学籍番号を必ず記載すること

for (初期化式; 条件式; インクリメント式)

for (i = 1 ; i <= 10 ; i++) {

処理

}

i++

インクリメント (i = i + 1)

i--

デクリメント (i = i - 1)

1から10までの数の和を求めるプログラム

```
#include<stdio.h>
int main(void)
{
    int sum, i;
    sum = 0;
    for ( i = 1; i < 11 ; i++)
    {
        sum = sum + i;
    }

    printf ( " 1から10までの和は%d ¥n " , sum );

    return 0;
}
```

i = 1	sum = sum + i (sum = 0 + 1 = 1)
i = 2	sum = sum + i (sum = 1 + 2 = 3)
i = 3	sum = sum + i (sum = 3 + 3 = 6)
:	
i = 10	sum = sum + i (sum = 45 + 10 = 55)

```
for ( i = 0 ; i < 10 ; i++ ) {
    処理
}
```

初期化式
条件式
インクリメント式
は自由に決定できる

```
for ( i = 10 ; i > 0 ; i-- ) {
    処理
}
```

```
for ( i = 0 ; i != 10 ; i++ ) {
    処理
}
```

例題1 (説明:スライド番号⑩まで)

1 の段から 9 の段までの九九を計算するプログラムを **for 文** を利用して作成してください。なお、実行結果が以下のようになるように、プログラムを作成すること。

＜実行結果＞

```

1 の段
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
-----
2 の段
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8

```

:

```

9 の段
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
-----
Press any key to continue

```

九九の1の段を表示するプログラム ※入力不要

例題1について
補足説明

```
#include<stdio.h>
int main(void)
{
    printf ( " %d × %d = %d¥n " , 1 , 1 , 1 *1 );
    printf ( " %d × %d = %d¥n " , 1 , 2 , 1 *2);
    printf ( " %d × %d = %d¥n " , 1 , 3 , 1 *3);
    printf ( " %d × %d = %d¥n " , 1 , 4 , 1 *4);
    printf ( " %d × %d = %d¥n " , 1 , 5 , 1 *5);
    printf ( " %d × %d = %d¥n " , 1 , 6 , 1 *6);
    printf ( " %d × %d = %d¥n " , 1 , 7 , 1 *7);
    printf ( " %d × %d = %d¥n " , 1 , 8 , 1 *8);
    printf ( " %d × %d = %d¥n " , 1 , 9 , 1 *9);

    return 0;
}
```

1 × 1 = 1
1 × 2 = 2
1 × 3 = 3
1 × 4 = 4
1 × 5 = 5
1 × 6 = 6
1 × 7 = 7
1 × 8 = 8
1 × 9 = 9

例題1について補足
説明
スライド⑦を書き直すと

for文を使って書き直すと・・・ ※入力不要

```
#include<stdio.h>
int main(void)
{
    int i;

    for(i=1; i<10; i++)
    {
        printf ( " %d × %d = %d¥n ", 1 , i , 1*i );
    }

    return 0;
}
```


※入力不要

9の段までの九九なので、
for文を使って書くと
以下のように9回までforループ
を書かないといけない

```
#include<stdio.h>
int main(void)
{
    int i, j;

    for(i=1; i<10; i++)
    {
        printf ( " %d × %d = %d¥n ", 1 , i , 1*i );

        for(i=1; i<10; i++)
        {
            printf ( " %d × %d = %d¥n ", 2 , i , 2*i );

            :
        }
        return 0;
    }
}
```

```
for(i=1; i<10; i++)
{
    printf ( " %d × %d = %d¥n ", 1 , i , 1*i );
}
```

1の段

```
for(i=1; i<10; i++)
{
    printf ( " %d × %d = %d¥n ", 2 , i , 2*i );
}
```

2の段

二重ループ 繰り返しの中で、さらに繰り返しを行うこともできる

そこで、二重ループを使い、
例題1を以下のように
書き直す

```
#include<stdio.h>
int main(void)
{
    int i, j;

    for (i = 1; i <= 9; i++)
    {
        printf("%d の段¥n", i);
        for (j=1; j<=9; j++)
        {
            printf("%d x %d = %d¥n", i, j, i * j);
        }
        printf("-----¥n", i);
    }

    return 0;
}
```

実際に各自のPCで入力し、
実行結果を確認してみ
てください

二重ループに関する例題

右図のように " * " を縦3行横5列に表示したい。
2重の for ループを使ってプログラムを作成せよ。
 画面表示には

```
printf( " * " ) ;  
printf( "¥n" ) ;
```



の2つ以外は使わないこと。

考え方

- 二重の for ループ
- *** を5個表示する部分を3回繰り返す**
 - * を5個表示するには
 - printf(" * ") ; で 「 * 」が一つだけ表示される
 - 次に printf(" * ") ; が実行されると「 * * 」になる
 ということは
 printf(" * "); を **5回繰り返せば** 「 * * * * * 」
 - ただし、次に printf(" * "); が実行されると * 6個になるので
 5回繰り返した後に改行 printf("¥n"); を行う

「*」を5個表示するため、書き方いろいろ

```
for ( i = 1 ; i <= 5 ; i++ )
{
    printf( "*" );
}
```

```
for ( i = 1 ; i <= 5 ; i++ )
    printf( "*" );
```

```
for ( i = 1 ; i < 6 ; i++ )
{
    printf( "*" );
}
```

```
for ( i = 1 ; i < 6 ; i++ )
    printf( "*" );
```

「*」を5個表示したら改行を入れる

```
for ( i = 1 ; i <= 5 ; i++ )
{
    printf( "*" );
}
printf( "¥n" ) ;
```

「*」を5個表示する部分を3回繰り返す

```
for ( j = 1 ; j <= 3 ; j++ )
{
    for ( i = 1 ; i <= 5 ; i++ )
    {
        printf( "*" );
    }
    printf( "¥n" ) ;
}
```

繰り返し処理 -while文-

```
while ( 条件式 )  
{  
    処理  
}
```

for (初期化式; 条件式; インクリメント式)

for 文 繰り返す回数の分かっているものに利用

while 文 実行してみないといつまで繰り返すかが分からないものに利用

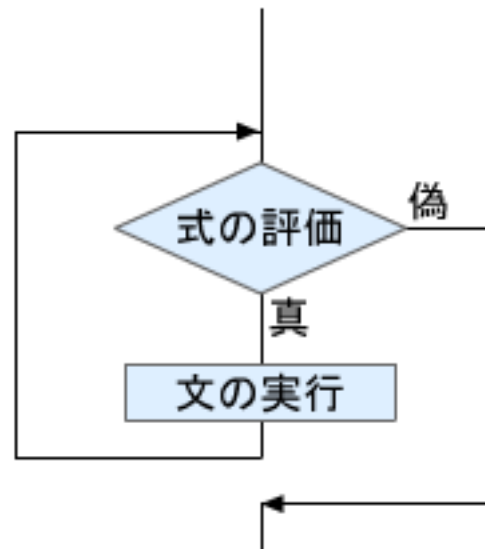
While文の構造

【「while文」の構造】

while(式)
文



for(; 式2 ;)
文



for 文 ループカウンタを使った繰り返す回数の分かっているものに利用

while 文 実行してみないと**いつまで繰り返すかが分からない**ものに利用

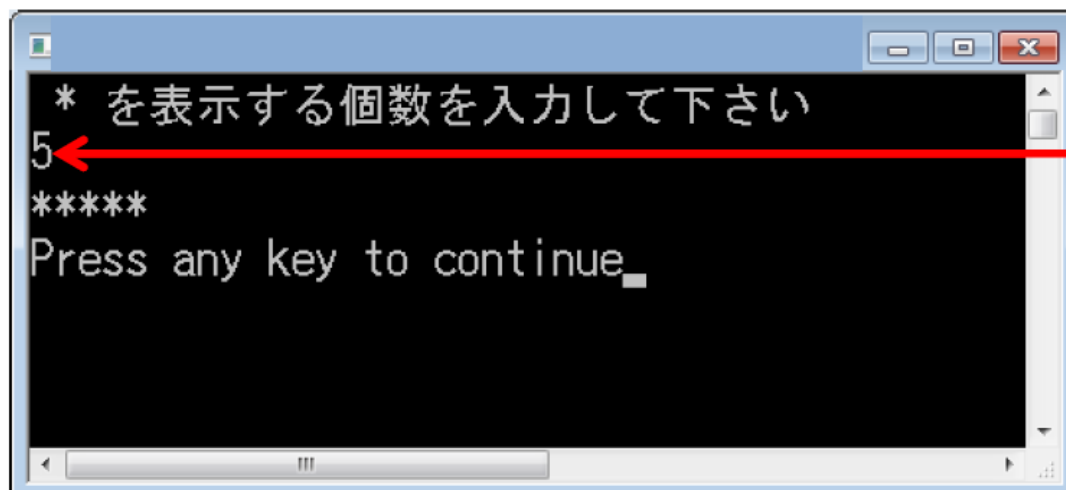
```
#include<stdio.h>
int main(void)
{
    int no ;
    scanf(" %d",&no);
    while ( no != 0 )
    {
        printf ( " %d¥n " , no);
        scanf( " %d ",&no);
    }

    return 0;
}
```


「キーボードから入力された値の回数だけ、画面に * を表示」
するプログラムです。

実行結果が以下のようになるように、プログラムを作成すること。

<実行結果>



```

* を表示する個数を入力して下さい
5
*****
Press any key to continue.
    
```

キーボードから
5と入力した場合

下記プログラムは、

「キーボードから入力された値の回数だけ、画面に * を表示」するプログラムです。

```
#include<stdio.h>
int main(void)
{
    int i=1,input;

    printf("* を表示する個数を入力して下さい¥n");
    scanf("%d",&input);

    for( i=1; i<=input; i++ )
    {
        printf( " * " );
    }
    printf("¥n");

    return 0;
}
```

実際に各自のPCで入力し、
実行結果を確認してみてください

while 文に書き換えたプログラム

「キーボードから入力された値の回数だけ、画面に * を表示」するプログラムです。

```
#include<stdio.h>
int main(void)
{
    int i=1,input;

    printf("* を表示する個数を入力して下さい¥n");
    scanf("%d",&input);

    while(i<=input){
        printf(" * ");
        i++;
    }

    printf("¥n");

    return 0;
}
```

実際に各自のPCで入力し、
実行結果を確認してみてください

前回のレポートに関する過去問

日付を入力すると、2016年2月の曜日を出力するプログラムを作成してください。なお、実行結果が以下のようにになるように、プログラムを作成すること。

SUN	MON	TUE	WED	THU	FRI	SAT
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29					

21 / 7 → 3...0 (日曜日)
 22 / 7 → 3...1 (月曜日)
 23 / 7 → 3...2 (火曜日)
 24 / 7 → 3...3 (水曜日)
 25 / 7 → 3...4 (木曜日)
 26 / 7 → 3...5 (金曜日)
 27 / 7 → 3...6 (土曜日)

Input day : 5
 Friday

考え方

- 整数変数 day を使って入力する
- (2月)1日は月曜日。
よって、day を7で割った余りで曜日 (youbi) が求められる

day % 7 ⇒
 1 : 月曜日
 2 : 火曜日
 3 : 水曜日
 4 : 木曜日
 5 : 金曜日
 6 : 土曜日
 0 : 日曜日

プログラム例(定義部、入力部)

```
#include <stdio.h>
int main( void )
{
    int  day, youbi ;

    printf( "Input day :" ) ;
    scanf( "%d", &day ) ;
```

プログラム例(比較、表示)

```
if ( ( day < 1 ) || ( day > 29 ) ) {  
    printf( "Input error! ¥n" ) ;  
}  
else {
```

```
    youbi = day % 7 ;  
    if ( youbi == 0 ) printf( "Sunday¥n" ) ;  
    if ( youbi == 1 ) printf( "Monday¥n" ) ;  
    if ( youbi == 2 ) printf( "Tuesday¥n" ) ;  
    if ( youbi == 3 ) printf( "Wednesday¥n" ) ;  
    if ( youbi == 4 ) printf( "Thursday¥n" ) ;  
    if ( youbi == 5 ) printf( "Friday¥n" ) ;  
    if ( youbi == 6 ) printf( "Saturday¥n" ) ;
```

```
}  
return 0 ;
```

```
}
```

例(1)

```
if ( youbi == 0 ) printf("Sunday¥n");  
else if ( youbi == 1 ) printf("Monday¥n");  
else if ( youbi == 2 ) printf("Tuesday¥n");  
else if ( youbi == 3 ) printf("Wednesday¥n") ;  
else if ( youbi == 4 ) printf("Thursday¥n") ;  
else if ( youbi == 5 ) printf("Friday¥n") ;  
else if ( youbi == 6 ) printf("Saturday¥n") ;
```

NG例

```
if ( youbi = 4 ) printf("Thursday¥n") ;
```

```
if ( youbi == 4 ) ;  
    printf("Thursday¥n") ;
```

```
if ( youbi != 4 ) { /* 論理が正しくない */  
    printf("Thursday¥n") ;  
}
```