

計算機プログラミング

第13回目

July 8, 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

■ ポインタ (スライド番号③～⑨)

■ 例題: 最も大きい値

- ポインタを使わない例 (スライド番号⑤)
- ポインタを使った例 (スライド番号⑥)

■ ファイルの入出力 (スライド番号⑩～⑪)

■ 前回(第12回目)レポート課題に関する解答例 (スライド番号⑫～⑯)

レポート提出あり

提出×切: 7月10日(土)20:00まで

- ポインタ - ポインタ型は
アドレスを表すデータ型の一つ

型 * 名前 ;

ポインタ変数を使うと...

ある変数の内容を間接的に参照・更新することができる

例: int * xyz ;

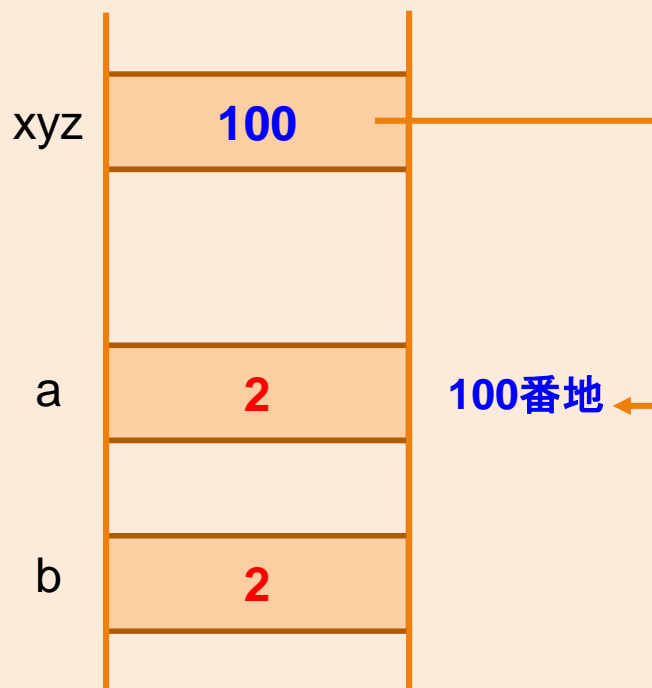
```
int * xyz;
int a, b;
```

```
xyz = &a;
```

```
* xyz = 2;
```

```
b = * xyz;
```

(& : 変数のアドレスを取り出す演算子)



- ポインタ -

例: `int * xyz;`

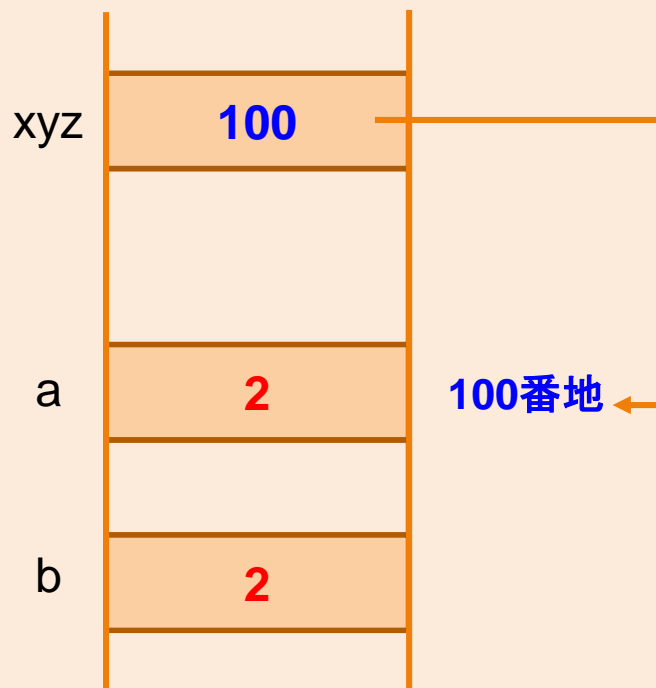
```
int * xyz;
int a, b;
```

```
xyz = &a;
```

```
* xyz = 2;
```

```
b = * xyz;
```

(&: 変数のアドレスを取り出す演算子)



変数に&演算子を適用すると、その変数の**アドレス(メモリの番地)**が取得できる

⇒ このアドレスのことを、その変数への**ポインタ**と呼ぶ

最も大きい値

(ポインタを使わない例)

```
#include<stdio.h>
int main(void)
{
    int  a, b, c, max;
    printf("3つの数字を入力:"); scanf("%d %d %d", &a, &b, &c);

    max = a;

    if( b > max){
        max = b;
    }
    if( c > max){
        max = c;
    }

    printf("最も大きい値は %d ", max);

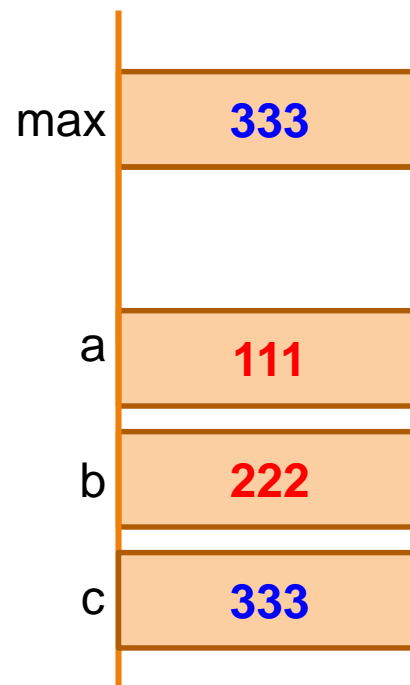
    return 0;
}
```

aに 111,
bに 222,
cに 333

この時点で、maxは
111となる

この時点で、maxは
222となる

この時点で、maxは
333となる



(ポインタを使った例)

```

:
int  a,b,c, * max;
printf("3つの数字を入力:"); scanf("%d %d %d", &a, &b, &c);

max = &a;

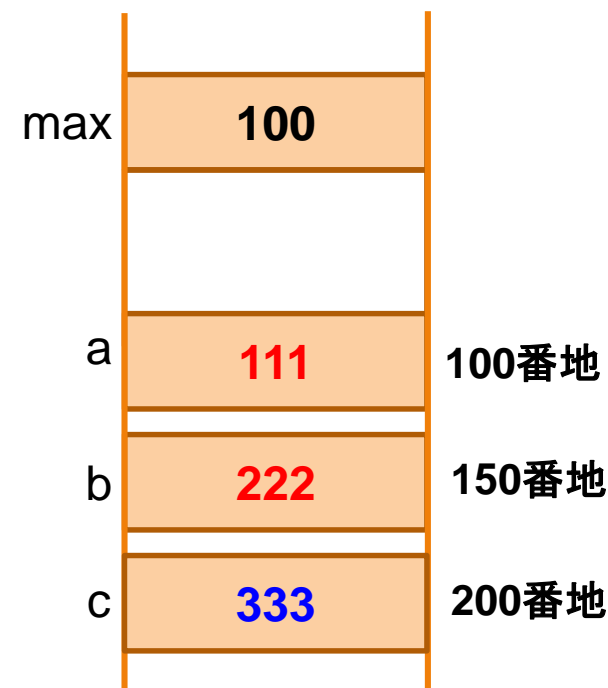
if( b > * max){
    max = &b;
}

if( c > * max){
    max = &c;
}

printf("最も大きい値は %d ", * max);
:
    
```

ポインタ変数

aのアドレスをmaxに代入



ポインタ変数を効果的に利用・・・整数値を加減算

```
int *point ;
```

```
point = point + 1 ;
```

```
point ++ ;
```

```
point += 2 ;
```

```
point -= 3 ;
```

現在指している位置から

1個先の変数を指すように移動

1個先の変数を指すように移動

2個先の変数を指すように移動

3個手前の変数を指すように移動

整数値を加減算することにより指す位置を移動することができる

ただし、

ポインタがさす変数はメモリ上連続していなければならない

ポインタ変数を効果的に利用・・・整数値を加減算

```
int ab[5] = {0}, * point ;
```

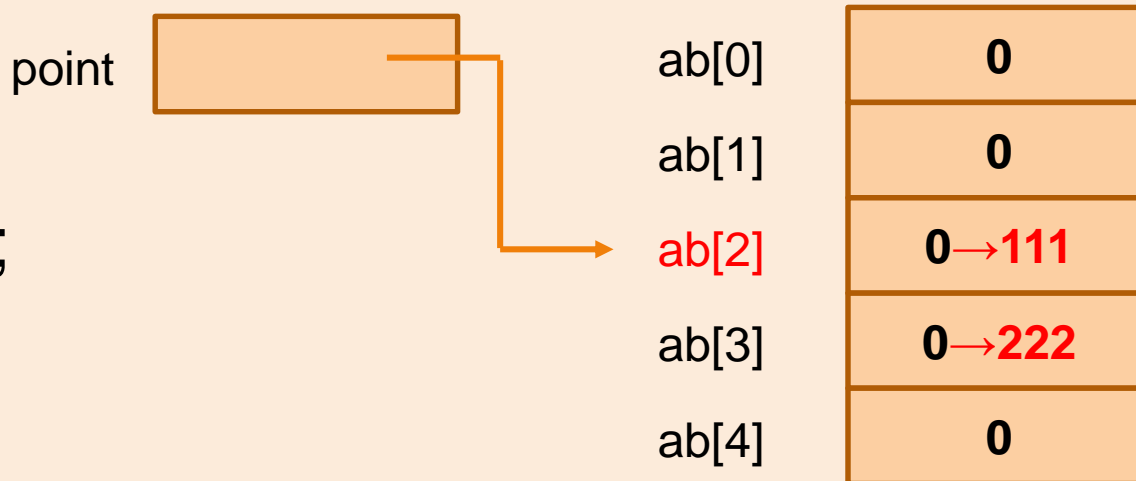
```
point = &ab[0] ;
```

```
point += 2 ;
```

```
* point = 111 ;
```

```
point ++ ;
```

```
* point = 222 ;
```




```
#include<stdio.h>
int main(void)
{
    int  tana[ ] = { 100,300,1000,500,200 };
    int i, s = 0, * point;

    point = &tana[0];

    for ( i=0; i<5; i++) {
        s += * point;
        point++;
    }

    printf("合計 : %d 円\n", s);

    return 0;
}
```

tana[0]

100

tana[1]

300

tana[2]

1000

tana[3]

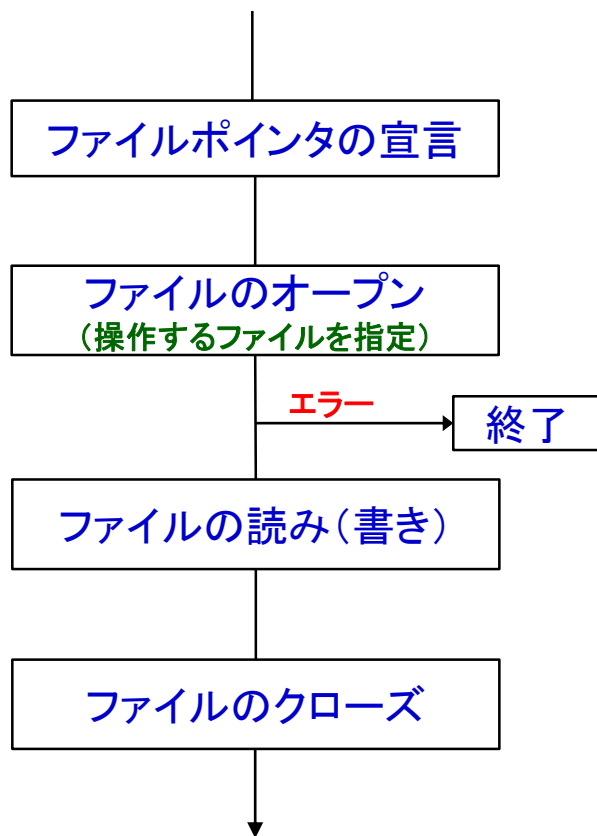
500

tana[4]

200

- ファイルの入出力 -

ハードディスク(HD)にあるデータを使ったり、HDに結果を記録するためには、ファイル(FILE)を読み書きできないといけない



- **ファイルポインタ**とは、FILE構造体へのポインタ
- FILE構造体は、入出力の現在位置、ファイルの終端に達したかの情報、エラー情報、関連するバッファへのポインタなどのファイルの入出力を行う上での必要不可欠な情報を管理

ファイル名を関連付けに行く

- ファイルの入出力 -

ファイル名(filename)で示されるファイルを、指定モード(mode)でオープンする。

モード	動作	ファイルがあるとき	ファイルがないとき
"r"	読み出し専用	正常	エラー(NULL返却)
"w"	書き込み専用	サイズを0にする(上書き)	新規作成
"a"	追加書きこみ専用	最後に追加する	新規作成

```
#include<stdio.h>
```

```
int main(void)
{
```

```
FILE* fp;
```

```
char moji[] = "software engineering";
```

```
int hoge = 100;
```

```
float foo = 153.5f;
```

```
fp = fopen("out_file.txt", "w");
```

```
fprintf(fp, "%s\n", moji);
```

```
fprintf(fp, "hoge is %d ", hoge);
```

```
fprintf(fp, "foo is %f\n", foo);
```

```
fclose(fp);
```

```
return 0;
```

```
}
```

ファイルポインタの宣言

ファイルのオープン

実際に書き込み

```
tnagai@golf-1[29]_ % gcc -o p7-1 p7-1.c
tnagai@golf-1[30]_ % ./p7-1
tnagai@golf-1[31]_ % cat out_file.txt
software engineering
hoge is 100 foo is 153.500000
tnagai@golf-1[32]_ %
```

ファイルのクローズ

問題1

第12回目のレポート
課題に関する解答例

リスト 1 は図 1 に示すような数字のピラミッドを表示するプログラムである。A~E の空欄を産めよ

```
#include <stdio.h>

int main(void) {
    int height, i, j;
    scanf("%d", &height);
    for (A; i <= height; i++) {
        for (j = 1; B; j++) {
            printf(C);
        }
        for (j = 1; j <= D; j++) {
            printf("%d", E);
        }
        printf("\n");
    }
    return 0;
}
```

リスト 1

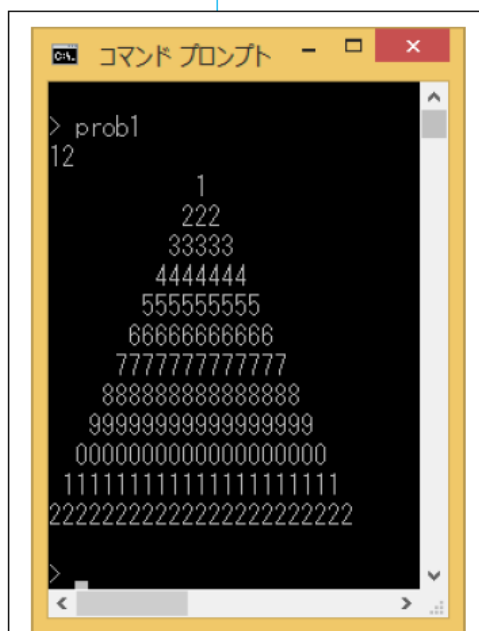


図 1 問題 1

問題1:解答例

第12回目のレポート
課題に関する解答例

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

    scanf("%d", &height);

    for (i = 1; i <= height; i++) {
        for (j = 1; j <= height - i; j++) {
            printf(" ");
        }
        for (j = 1; j <= i * 2 - 1; j++) {
            printf("%d", i % 10 );
        }
        printf("\n");
    }
    return 0;
}
```

A:

i = 1

B:

j <= height - i

C:

" "

D:

i * 2 - 1

E:

i % 10

問題2 図2に示すような図形を表示するプログラムを作成する。図形の左には星の数を表示し、最後にすべての星の数を表示するプログラムを作成せよ。
(#include から始めてください。#include <stdlib.h>も追加すること) (20 点)



図2 問題2

考え方

- ある1行を考えると

- k 個の空白
- m 個の "*"

- 空白の個数 k 3, 2, 1, 0, 1, 2, 3

$$k = \text{abs}(4-i) \text{ or } k = \text{abs}(i-4)$$

あるいは

$$k = 4-i ;$$

$$\text{if } (k < 0) \{ k = -k ; \}$$

- 星の数 m 1, 3, 5, 7, 5, 3, 1

$$k + m + k = 7 \text{ なので } m = 7 - 2 * k$$

問題2 解答例

第12回目のレポート
課題に関する解答例

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i, j, k, m, sum = 0;
    for ( i = 1 ; i <= 7 ; i++ )
    {
        k = abs( i - 4 ) ;
        m = 7 - 2 * k ;

        printf("%d", m);

        sum = sum + m ;

        for ( j = 1 ; j <= k ; j++ ){
            printf( " " ) ;
        }
        for ( j = 1 ; j <= m ; j++ ){
            printf( "*" ) ;
        }
        printf("¥n") ;
    }
    printf("星の数 %d", sum);
    return 0;
}
```

(定義部＋
空白と星の数)

表示

問題 3 以下の整数配列 `c0[7]` は 7bit の数値である。このデータに CRC 操作 (左ローテーション) を行い、結果を 10 進数表示するプログラムを作成せよ。

	0	1	2	3	4	5	6
<code>c0[7]</code>	0	0	1	1	0	1	0

操作後

	0	1	2	3	4	5	6
<code>c1[7]</code>	0	1	1	0	1	0	0

第12回目のレポート
課題に関する解答例

```
#include <stdio.h>

int main(void) {

    int    c0[7] = {0,0,1,1,0,1,0};
    int    c1[7];

    int    i, tmp, dec ;

    c1[6] = c0[0];

    for (i = 0; i <= 5; i++) {
        c1[i] = c0[i + 1];
    }
```

例2: 10進数変換

```
dec = 0;
tmp = 1;

for (i = 6; i >= 0; i--) {
    dec = dec + tmp * c1[ i ];
    tmp = tmp * 2 ;
}
```

この部分は
どちらでもいける

例3: 10進数変換

```
dec = 0;
tmp = 64; // 2^6

for (i = 0; i <= 6; i++) {
    dec = dec + tmp * c1[ i ];
    tmp = tmp / 2 ;
}
```

```
dec = c1[0];

for (i = 1; i <= 6; i++) {
    dec *= 2;
    dec += c1[i];
}

printf("10進数 : %d\n", dec);
return 0;
```


<本日のレポート>

【問題 1】 下に示すような 3 行 3 列の `int` の行列 `A` がある。

この行列に対して要素が左 90 度回転した行列 `B` に変換し表示するプログラムを作成せよ。

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$B = \begin{pmatrix} 3 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{pmatrix}$$

(`#include` から始めてください)

【問題 2】

右に示すように、縦横にタイトルのある九九の表を作成せよ。

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18

(以下省略)

【問題 3】

配列に予め初期値として 7, 4, 12, 71, 3, 85, 69, 47, 11 の **9個** を設定しておき、この中で **素数** のものを表示するプログラムを作成せよ。

例えば、7 が素数かどうか調べる場合...

7 ÷ 2, 7 ÷ 3, 7 ÷ 4, 7 ÷ 5, 7 ÷ 6 の余りを調べる

↑

2 から目的の数値より 1 小さい値で除算を行い、余りが 0 になるかどうかで分かります。

一度でも余りがゼロになれば、目的の数値は素数ではありません。

```
#include<stdio.h>
int main(void)
{
    int data[9] = _____
    int i,j;
    printf("素数 : ");
    for(i=0; i<9; i++){
        for(_____) {
```

【問題 4】 並べ替え(バブルソート)

与えられた整数を小さい順に並び替えるバブルソートについて今回学びます。
具体的にどのようなものかと言いますと、今、

37 23 11 19 5 7

という6個の整数が与えられたとき、これを小さい順に並び替えて、

5 7 11 19 23 37

にするということです。

実際、どのようなアルゴリズムをするのでしょうか？

考え方

- (1) まず一番目の37を2番目の23と比較する。 $37 > 23$ より、これを取りかえる。
- (2) 次に2番目の整数(今の場合は37)と3番目の整数を比較し、3番目の整数が小さかったら取り替える。

- (3) 以下これを繰り返すと、

23 11 19 5 7 37

となります。

- (4) 次に、上の操作(1) - (3)を前から5つの整数に行う。そうすると、

11 19 5 7 23 37

となります。

- (5) 以下、順に比較する整数を減らして、(1) - (3)の操作を行う。

つまり、比較する整数の集合を配列 $d[n]$ であらわしたとき、上のアルゴリズムは、

```
for(i = n-1; i >= 0; i--)
    {d[0] と d[2], ..., d[i] を比較し、最大数を d[i] とする}
```

となります。

それでは実際に組んでみましょう。