# プログラミング言語Ⅱ

第2回レポート

学籍番号:5535089U

氏名:中山颯

# レポート課題7-4

• 以下のプログラムを実行した際のmain関数内の動作を一行ずつ解説 せよ。(例題を参考に、次ページのスライドをコピーして記入すること)

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
07: scanf("%d", pt);
08: if (*pt min > *pt) {
09:
          pt min = pt;
10:
11:
       sum += *pt;
12:
        pt++;
13:
14:
     sum -= *pt min;
     printf("sum = %d\u00e4n", sum);
15:
16:
     return 0;
17: }
```

```
実行 :
./a.out (enter)
78 11 90 (enter)
sum = 168
```

注意:メモリサイズは下記の設定とする int: 4バイト

float: 4バイト

double: 8バイト

ポインタ変数:8バイト

また、アドレスの最初は1000番地とする

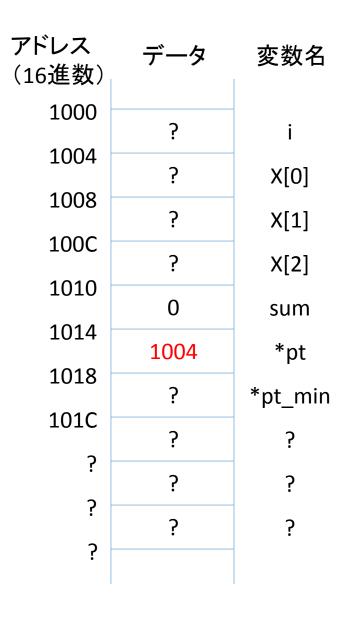
- int型変数 iをメモリ上に確保する
- int型の1次元配列x(要素数3)をメモリ上に確保する
- int型変数sumをメモリ上に確保し0を代入する
- int型のポインタ変数\*pt , \*pt\_minをメモリ上に確保する

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
07: scanf("%d", pt);
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
       sum += *pt;
11:
12:
       pt++;
13:
14: sum -= *pt min;
     printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```

アドレス (16進数)	データ	変数名
1000	?	i
1004	?	X[0]
1008	?	X[1]
100C	?	X[2]
1010 1014	0	sum
	?	*pt
1018	?	*pt_min
101C	?	?
j	?	?
?	?	<b>;</b>
<b>;</b>		

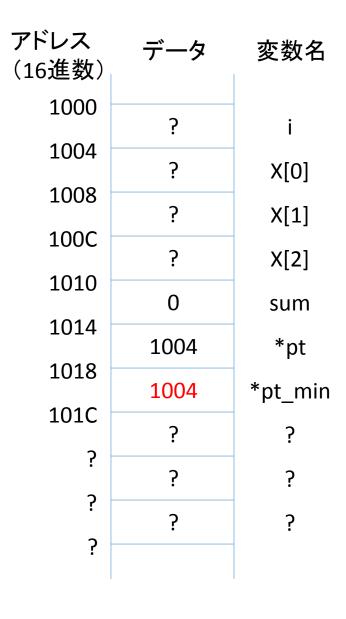
• ptに一次元配列xの先頭のアドレスを代入する

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt_min;
     printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



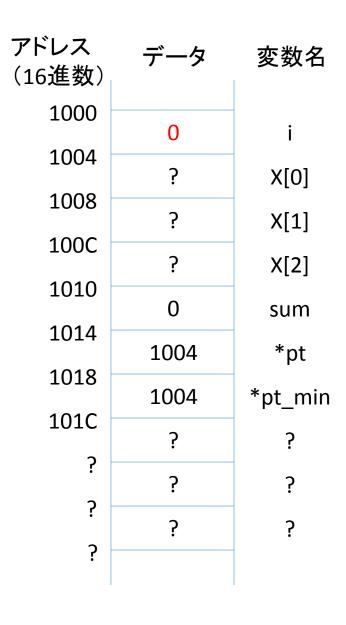
• pt\_minにptの値をコピーする

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
07:
       scanf("%d", pt);
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt_min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



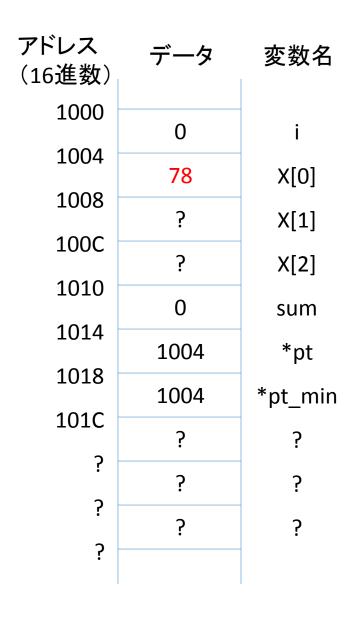
- forループ1回目
- 初期化式が実行され、iに0が代入される
- 継続条件式が評価され、i < 3 という条件を満たしているため継続される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
          pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



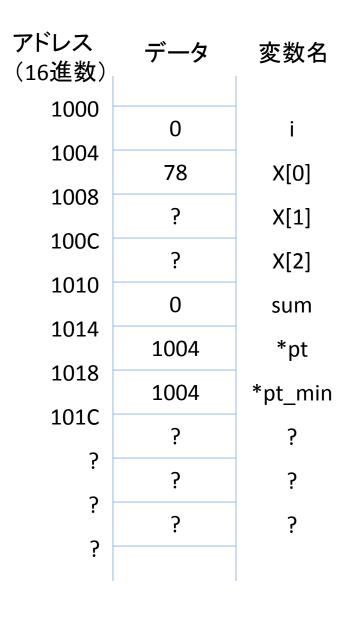
- scanf関数で、標準入力から入力された数値をアドレスptの場所に格納する
- 現在ptはx[0]のアドレスを指しているため、x[0]に78が格納される

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



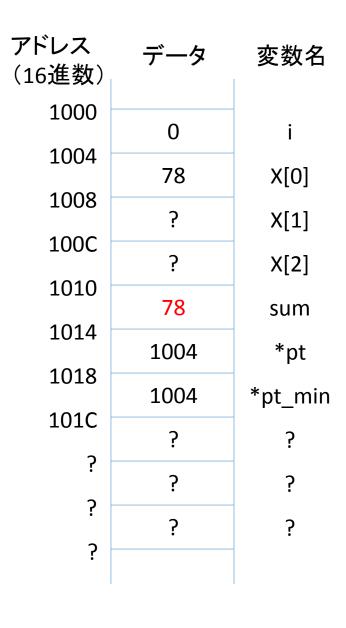
- \*ptと\*pt\_minに入っている値を比較する. この場合 \*pt=78,\*pt\_min=78を指す.
- 同じ大きさなのでif文の中の処理を行わずに処理を進める.

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



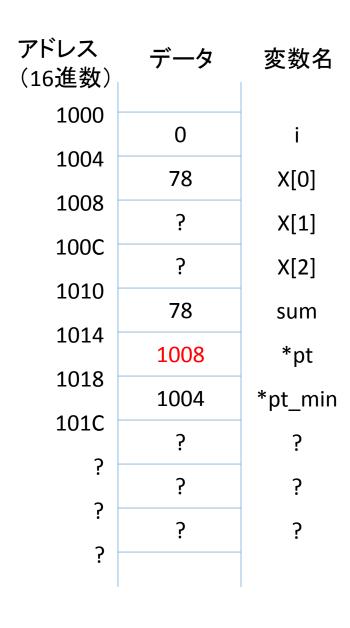
- 変数sumにptが指している場所の中身を加算する
- 現在ptはx[0]のアドレスを指しているため、x[0]の値78 が加算される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



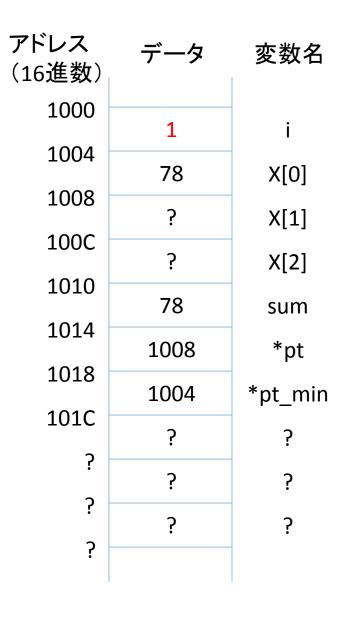
- ポインタ変数ptをインクリメントし、次の要素のアドレスを指すようになる
- 現在ptはx[0]のアドレスを指しているため、x[1]のアドレスが格納される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
          pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



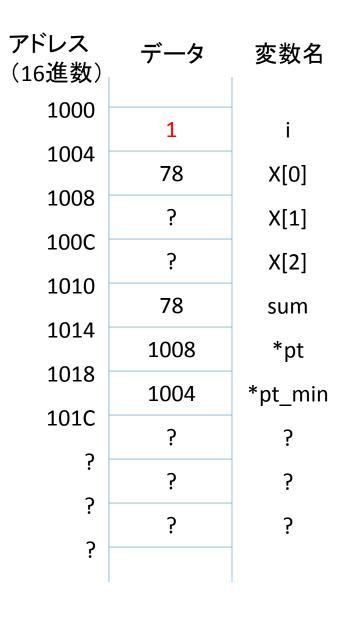
- 1回目のforループが終了
- 再初期化式が実行され、iがインクリメントされる
- 6行目に戻る

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
    pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



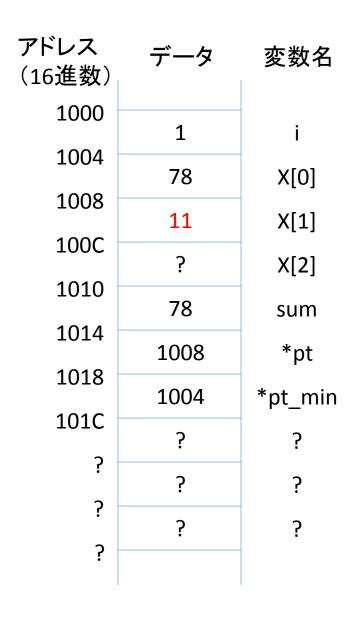
- forループ2回目
- 継続条件式が評価され、i < 3 という条件を満たしているため継続される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
     return 0;
16:
17: }
```



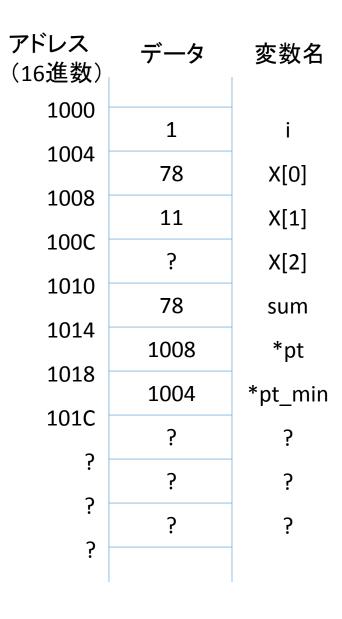
- scanf関数で、標準入力から入力された数値をアドレスptの場所に格納する
- 現在ptはx[1]のアドレスを指しているため、x[1]に11が格納される

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



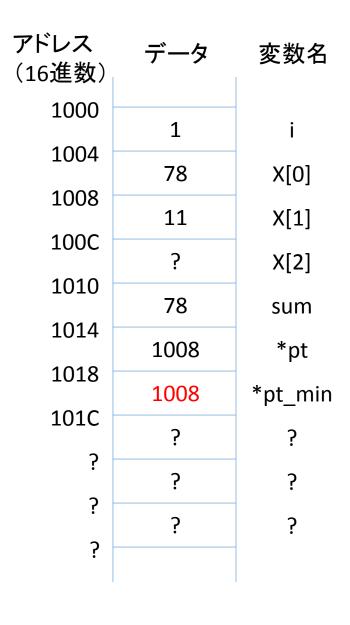
- \*ptと\*pt\_minに入っている値を比較する. この場合 \*pt=11,\*pt\_min=78を指す.
- \*ptの方が小さいのでif文の中の処理を行う.

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
    sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



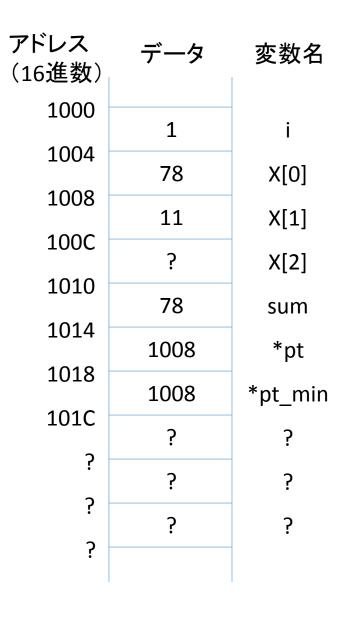
\*pt\_minに\*ptの値を代入する. つまりx[1]のアドレスを\*pt\_minに 渡す.

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



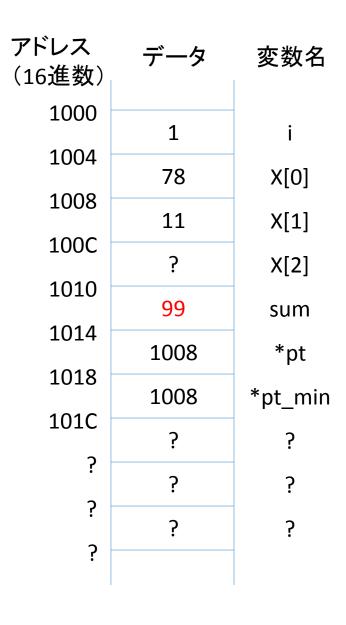
• if文の中の処理終了次の行の処理に移る.

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
07:
       scanf("%d", pt);
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt_min;
      printf("sum = %d\u00e4n", sum);
     return 0;
16:
17: }
```



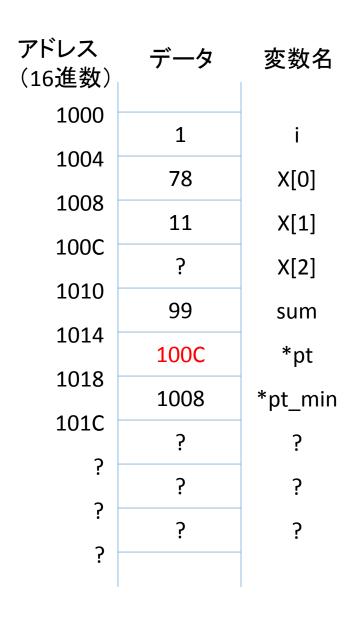
- 変数sumにptが指している場所の中身を加算する
- 現在ptはx[1]のアドレスを指しているため、x[1]の値11が加算される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



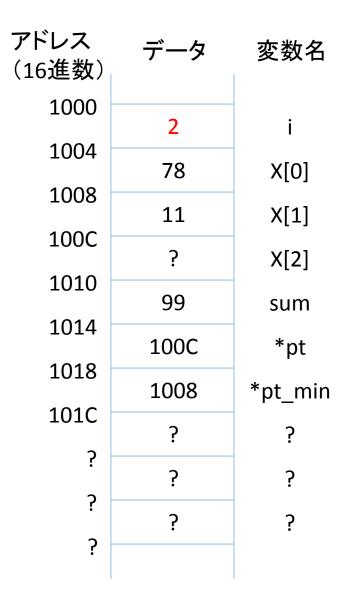
- ポインタ変数ptをインクリメントし、次の要素のアドレスを指すようになる
- 現在ptはx[1]のアドレスを指しているため、x[2]のアドレスが格 納される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
          pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



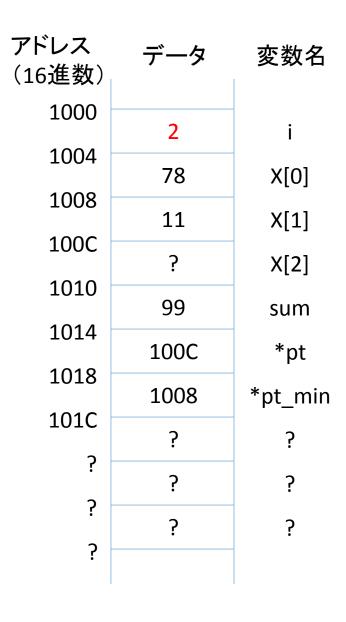
- 2回目のforループが終了
- 再初期化式が実行され、iがインクリメントされる
- 6行目に戻る

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



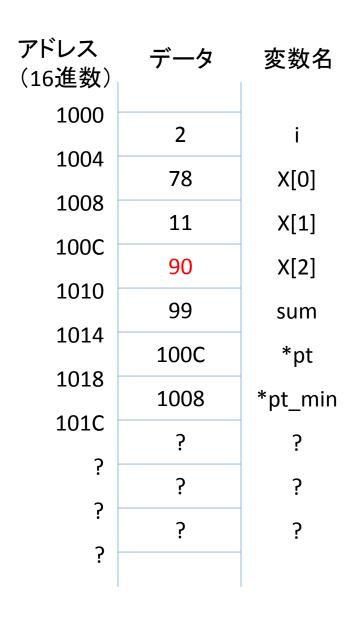
- forループ3回目
- 継続条件式が評価され、i < 3 という条件を満たしているため継続される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
     return 0;
16:
17: }
```



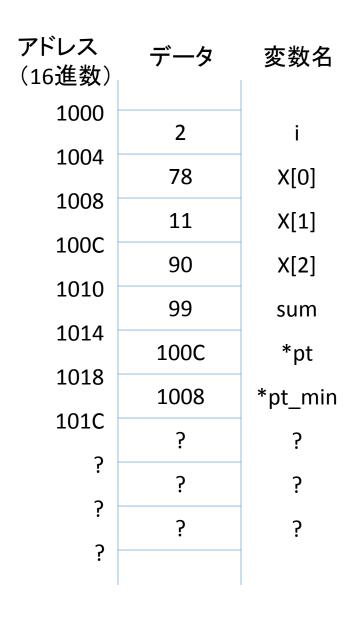
- scanf関数で、標準入力から入力された数値をアドレスptの場所に格納する
- 現在ptはx[3]のアドレスを指しているため、x[3]に90が格納される

```
01: #include <stdio.h>
02: int main() {
    int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



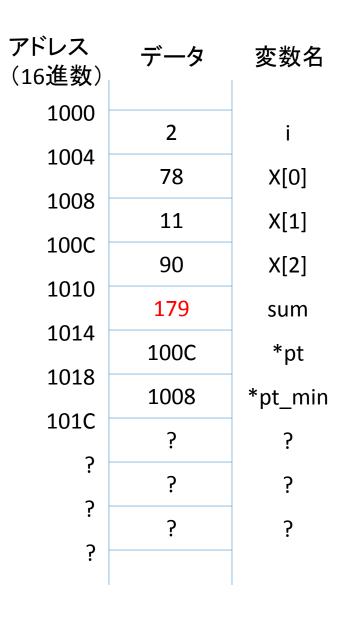
- \*ptと\*pt\_minに入っている値を比較する. この場合 \*pt=90,\*pt\_min=11を指す.
- \*pt\_minの方が大きいのでif文の中の処理を行わずに11行目に 移動.

```
01: #include <stdio.h>
02: int main() {
03: int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



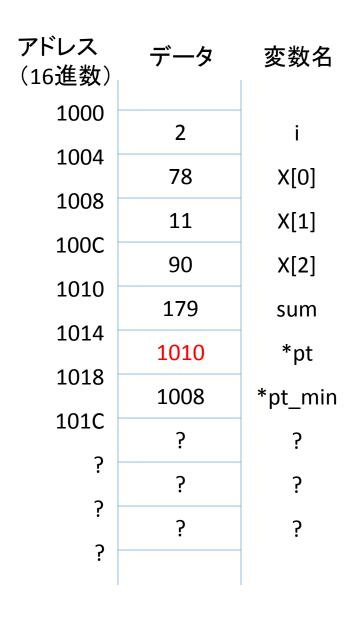
- 変数sumにptが指している場所の中身を加算する
- 現在ptはx[3]のアドレスを指しているため、x[3]の値90が加算される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



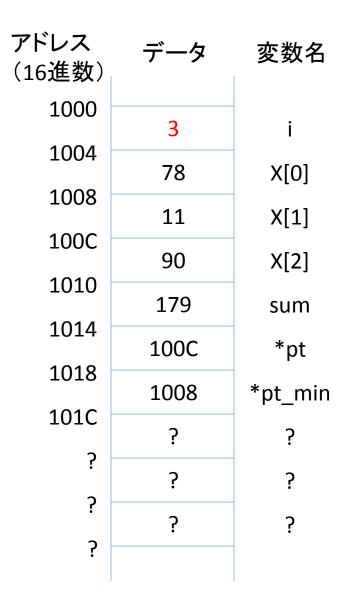
- ポインタ変数ptをインクリメントし、次の要素のアドレスを指すようになる
- 現在ptはx[3]のアドレスを指しているため、sumのアドレスが格納される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
          pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



- 3回目のforループが終了
- 再初期化式が実行され、iがインクリメントされる
- 6行目に戻る

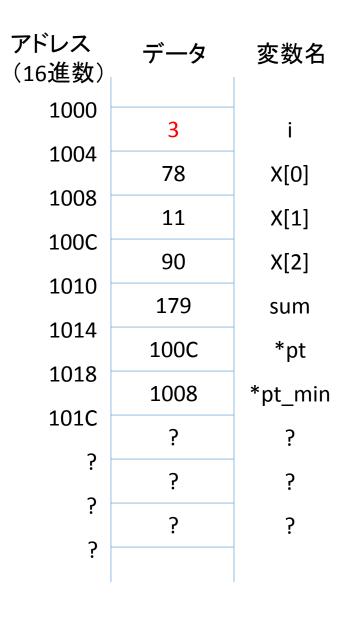
```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



#### 6行目(4回目)

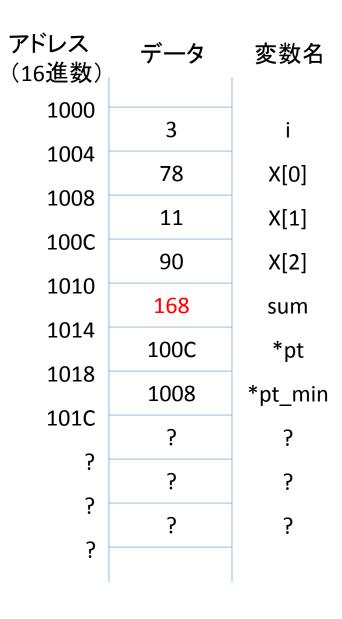
- forルーフを3回回し終わった段階で右の状態
- forルーフ4回目では、継続条件式が評価され、i < 3 という条件を満たしていないためループを抜けて14行目に進む

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt min;
04: pt = x;
05: pt min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt min > *pt) {
08:
09:
         pt min = pt;
10:
        sum += *pt;
11:
12:
        pt++;
13:
14: sum -= *pt min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



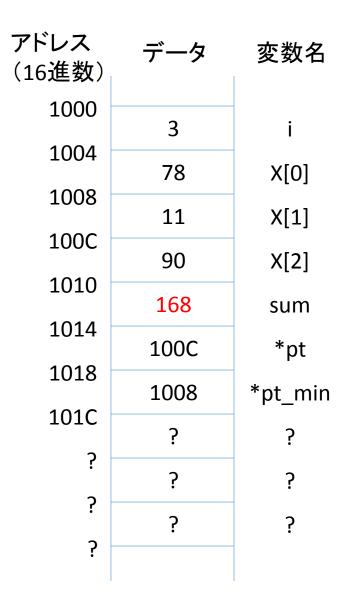
• 変数sumから\*pt\_minが指す値, つまり11を引く

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt_min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



- printfでsumの内容を表示する
- 標準出力に"sum = 168"と表示される

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
          pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
     sum -= *pt_min;
      printf("sum = %d\u00e4n", sum);
16:
     return 0;
17: }
```



• 関数の戻り値として0を返し、main関数を終了する

```
01: #include <stdio.h>
02: int main() {
     int i, x[3], sum=0, *pt, *pt_min;
04:
     pt = x;
05: pt_min = pt;
06: for (i=0; i<3; i++) {
       scanf("%d", pt);
07:
       if (*pt_min > *pt) {
08:
09:
         pt min = pt;
10:
11:
        sum += *pt;
12:
        pt++;
13:
14: sum -= *pt_min;
      printf("sum = %d\u00e4n", sum);
     return 0;
16:
17: }
```

