

第4回 C言語勉強会

問題演習 そのいち

丸山翔平

問題 1

1. 設問

2. 解き方のヒント

3. 参考に...

4. 解答例

1. 設問

●基礎—数値の大小判断

- ふたつの数字のうち、どちらが大きいかを判定する

差があるとき

```
一つ目の数値を入力してください  
>>15  
二つ目の数値を入力してください  
>>6  
15は6より大きいです
```

差がないとき

```
一つ目の数値を入力してください  
>>17  
二つ目の数値を入力してください  
>>17  
ふたつの数値は等しいです
```

●応用—文字の前後判断

- ふたつのアルファベットのうち、どちらが最初に来るかを判定する.

```
一つ目の文字を入力してください  
>>D  
二つ目の文字を入力してください  
>>J  
DはJより先に来ます
```

```
一つ目の文字を入力してください  
>>Y  
二つ目の文字を入力してください  
>>Y  
異なる文字を入力してください
```

2. 解き方のヒント

●使うC言語の構文は…？

• キーボードからの受取り：

• 条件分岐をする：

• 画面に結果を表示する：

• 変数はいくつ必要か？

• 入力を受け取る分だけ→

• いくつに分岐させる必要があるか？

• $c1$ が $c2$ より大きいとき

• $c2$ が $c1$ より大きいとき

• どちらも同じとき

• 予期しない入力があったとき

3. 参考に...

- 文字のデータの中身

ASCII文字コード (文字に番号を割り当てたもの)

ASCII文字コード

文 字	10 進	16 進																					
NUL	0	00	DLE	16	10	SP	32	20	@	64	40	P	80	50	`	96	60	p	112	70			
SOH	1	01	DC1	17	11	!	33	21	A	65	41	Q	81	51	a	97	61	q	113	71			
STX	2	02	DC2	18	12	"	34	22	B	66	42	R	82	52	b	98	62	r	114	72			
ETX	3	03	DC3	19	13	#	35	23	C	67	43	S	83	53	c	99	63	s	115	73			
EOT	4	04	DC4	20	14	\$	36	24	D	68	44	T	84	54	d	100	64	t	116	74			
ENQ	5	05	NAK	21	15	%	37	25	E	69	45	U	85	55	e	101	65	u	117	75			
ACK	6	06	SYN	22	16	&	38	26	F	70	46	V	86	56	f	102	66	v	118	76			
BEL	7	07	ETB	23	17	'	39	27	G	71	47	W	87	57	g	103	67	w	119	77			
BS	8	08	CAN	24	18	(40	28	H	72	48	X	88	58	h	104	68	x	120	78			
HT	9	09	EM	25	19)	41	29	I	73	49	Y	89	59	i	105	69	y	121	79			
LF*	10	0a	SUB	26	1a	*	42	2a	:	58	3a	J	74	4a	Z	90	5a	j	106	6a	z	122	7a
VT	11	0b	ESC	27	1b	+	43	2b	;	59	3b	K	75	4b	[91	5b	k	107	6b	{	123	7b
FF*	12	0c	FS	28	1c	,	44	2c	<	60	3c	L	76	4c	\	92	5c	l	108	6c		124	7c
CR	13	0d	GS	29	1d	-	45	2d	=	61	3d	M	77	4d]	93	5d	m	109	6d	}	125	7d
SO	14	0e	RS	30	1e	.	46	2e	>	62	3e	N	78	4e	^	94	5e	n	110	6e	~	126	7e
SI	15	0f	US	31	1f	/	47	2f	?	63	3f	O	79	4f	_	95	5f	o	111	6f	DEL	127	7f

%cで格納した中身はこんな感じ



簡単に解けちゃった人は

- 数値の大小を判断して，その差も一緒に表示

```
一つ目の数値を入力してください  
>>15  
二つ目の数値を入力してください  
>>6  
15は6より9だけ大きいです
```

- アルファベットの前後を判断して，その間に
来るアルファベットをすべて表示

(要 繰り返し構文)

```
一つ目の文字を入力してください  
>>D  
二つ目の文字を入力してください  
>>J  
DはJより先に来ます  
...DEFHIJ...
```



4. 解答例

基礎—数値の大小判定

```
#include <stdio.h>

int main(void){
    int c1, c2;
    printf("一つ目の数値を入力してね¥n");
    scanf("%d", &c1);
    printf("二つ目の数値を入力してね¥n");
    scanf("%d", &c2);

    if(c1 > c2)
        printf("%d は %d より大きい¥n", c1, c2);
    else if(c1 < c2)
        printf("%d は %d より大きい¥n", c2, c1);
    else if(c1 == c2)
        printf("%d と %d は同じです¥n", c1, c2);
    else
        printf("error.¥n");

    return 0;
}
```

基礎—数値の大小判定

引数をもう一つとって, 計算させてやればOK

```
if(c1 > c2)
    printf("%d は %d より %d だけ大きい¥n", c1, c2, c1 - c2);
else if(c1 < c2)
    printf("%d は %d より %d だけ大きい¥n", c2, c1, c2 - c2);
else if(c1 == c2)
    printf("%d と %d は同じです¥n", c1, c2);
else
    printf("error.¥n");
```

応用一文字の前後判定

```
#include <stdio.h>

//おまじない
void ArrangeNum(char c1, char c2);

int main(){
    char c1, c2, buf;
    printf("please enter first character.¥n");
    scanf("%c", &c1);
    buf = getchar(); //これがないと二文字目をスルーする
    printf("enter second character.¥n");
    scanf("%c", &c2);

    //大文字小文字の判定
    if('a' <= c1 && c1 <= 'z' && 'A' <= c2 && c2 <= 'Z')
        c1 -= 32;
    else if('A' <= c1 && c1 <= 'Z' && 'a' <= c2 && c2 <= 'z')
        c2 -= 32;
```

```
if(c1 < c2) {
    printf("%c appears before %c¥n", c1, c2);
    ArrangeNum(c1, c2);
}
else if(c1 > c2){
    printf("%c appears after %c¥n", c1, c2);
    ArrangeNum(c2, c1);
}
else if(c1 == c2)
    printf("%c and %c are the same¥n", c1, c2);
else
    printf("error.¥n");
return 0;
}

//文字を羅列する"関数"
void ArrangeNum(char c1, char c2){
    int i;
    printf("...");
    for(i = c1; i <= c2; i++){
        printf("%c", i);
    }
    printf("...¥n");
}
```

問題演習 ②

安藤

課題 (for文を用いて)

- 1から50の間にある奇数を全て足して出来る数を求めてみよう。

回答例

```
#include <stdio.h>↓
↓
int main()↓
{↓
    int i;↓
    int wa = 0;↓
    for(i=1;i<=50;i=i+2)↓
    {↓
        wa=wa+i;↓
        ↓
        printf("%d ¥n",wa);↓
        ↓
    }↓
    return 0;↓
}↓
```

問題演習 ③

佐藤

for文を使った応用問題

- 1 ~ 100までの数字を表の様に出力したい

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	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

1) 左上から右下に横向き

100	99	98	97	96	95	94	93	92	91
90	89	88	87	86	85	84	83	82	81
80	79	78	77	76	75	74	73	72	71
70	69	68	67	66	65	64	63	62	61
60	59	58	57	56	55	54	53	52	51
50	49	48	47	46	45	44	43	42	41
40	39	38	37	36	35	34	33	32	31
30	29	28	27	26	25	24	23	22	21
20	19	18	17	16	15	14	13	12	11
10	9	8	7	6	5	4	3	2	1

2) 右下から左上に横向き

91	81	71	61	51	41	31	21	11	1
92	82	72	62	52	42	32	22	12	2
93	83	73	63	53	43	33	23	13	3
94	84	74	64	54	44	34	24	14	4
95	85	75	65	55	45	35	25	15	5
96	86	76	66	56	46	36	26	16	6
97	87	77	67	57	47	37	27	17	7
98	88	78	68	58	48	38	28	18	8
99	89	79	69	59	49	39	29	19	9
100	90	80	70	60	50	40	30	20	10

3) 右上から左下に縦向き

それぞれについて
for文を工夫して
結果を出力しよう

for文を使った応用問題：ヒント

- 1)をどう書けばいいか？

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	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

上の方は“行”を表す

```
for ( [ ] ; [ ] ; [ ] ){  
  for ( [ ] ; [ ] ; [ ] ){
```

下の方は“列”を表す

1行目は $0+1$, $0+2$, ... $0+10$
2行目は $10+1$, $10+2$, ... $10+10$

規則性を式に取り入れる

- 入れ子構造が使えるそう
→for文の中にfor文を書く
- $1 \sim 10$, $11 \sim 20$...
→規則性がある

for文を使った応用問題： 1)

- まずは1) を問いて見よう

```
int main(){
int x, y;
for ( [redacted]; [redacted]; [redacted] ){
for ( [redacted]; [redacted]; [redacted] ){
printf("%3d", [redacted])
}
[redacted]
}
return 0;
}
```

for文を使った応用問題：1) 答

- まずは1) を問いて見よう

```
int main(){
  int x, y;
  for (y = 1; y <= 10; y++){
    for (x = 1; x <= 10; x++){
      printf("%3d", (y - 1) * 10 + x);
    }
    printf("¥n");
  }
  return 0;
}
```

出発点をここで変える

横向きか縦向きか決める

10個おきに改行

- これを少し変えて2) と3) を問いてみよう

応用問題：答 2)、3)

● 2) の解答

```
for (y = 10; y >= 1; y--){  
    for (x = 10; x >= 1; x--){  
        printf("%3d", (y - 1) * 10 + x);  
    }  
    printf("¥n");  
}
```

for (初期値、範囲、次)
中身を1) と逆にする

● 3) の解答

```
for (y = 1; y <= 10; y++){  
    for (x = 10; x >= 1; x--){  
        printf("%3d", (x - 1) * 10 + y);  
    }  
    printf("¥n");  
}
```

列(x)部分を1) と逆に
式の中身のx, yを逆に

問題演習 ④

コラッツの問題

加納 徹

コラッツの問題

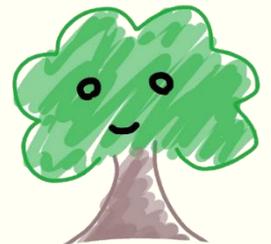
任意の自然数 n に対して、以下の操作を繰り返す。

$\left\{ \begin{array}{l} n \text{ が偶数の場合、} n \text{ を } 2 \text{ で割る。} \\ n \text{ が奇数の場合、} n \text{ に } 3 \text{ をかけて } 1 \text{ を足す。} \end{array} \right.$

コラッツ予想：

上述の操作に対して、どのような初期値 n を与えても、有限回の試行で $1 \rightarrow 4 \rightarrow 2 \rightarrow 1$ のループに陥る。

数論における未解決問題



while文

条件式を満たす間、繰り返しを行う文法

- 繰り返し回数がわからないとき
- 特別な条件で終了させたいとき

```
while (条件式) {  
    繰り返したい内容;  
}
```

for ループよりもすっきりさせることができる



while文

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

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

```
    int sum = 0, n;
```

```
    while (sum <= 10) {
```

```
        printf("数値を入力して下さい。¥n");
```

```
        scanf("%d", &n);
```

```
        sum = sum + n;
```

```
        printf("現在の値は %d です。¥n", sum);
```

```
    }
```

```
    printf("合計が 10 を超えました。¥n");
```

```
    return 0;
```

```
}
```

sum が 10 よりも小さい間、繰り返す

入力された数値を加えていく

break; と書いてループから抜けることもできる



問題

問題 1

コラッツの問題において、50 未満の自然数 n が 1 に到達するまでに要したステップ数を表示し、最もステップ数が多かった数字を調べなさい。

問題 2

100 万未満の自然数 n について、検証しなさい。

問題 1 のヒントは次のページ



ソースコード例

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

```
int main(void) {  
    int n, count = 0;
```

ステップ数を数えるための変数

```
    printf("n の値を入力して下さい。¥n");  
    scanf("%d", &n);
```

```
    while ( (a) nが1でない間繰り返す
```

```
        if (n % 2 == 0) {
```

```
            (b) 偶数なら2で割る
```

```
        } else {
```

```
            (c) 奇数なら3を乗じて1を足す
```

```
        }
```

```
        count++;
```

```
    }
```

```
    printf("ステップ数は %d です。¥n", count);
```

```
    return 0;
```

```
}
```



終わり
