条件判断 switch 文

if 文は 2 分岐の条件判断。多分岐の条件判断には switch 文を用いる

```
switch(式){
  case 定数1 : 文1; break;
  case 定数2 : 文2; break;
  ...
  case 定数n : 文n; break;
  default : 文n+1; break;
}
```

式は整数型もしくは文字型

case 定数 : をラベル(名札)という

式を評価し、その値が定数であるラベルの文へ処理が移る。break 文に出会うと switch 文を終了

式の値に合致する定数が無ければ default ラベルの文に処理が移る(例外処理)

switch 文は、式の値によって処理を複数に分岐

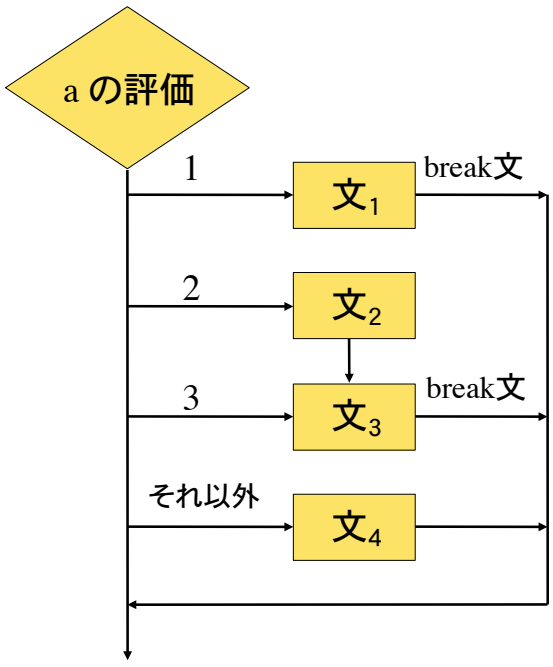
文の後の break 文は無くても良い(無いと分岐後の処理手順が異なる)

switch 文と break 文

switch 文は、式の値に対応するラベルへ処理を分岐させる。
break 文は、その文が書かれた処理を終了させる(この場合 switch 文)

```
int a;
scanf("%d", &a);

switch(a){
  case 1 : 文1; break;
  case 2 : 文2;
  case 3 : 文3; break;
  default : 文4;
}
```



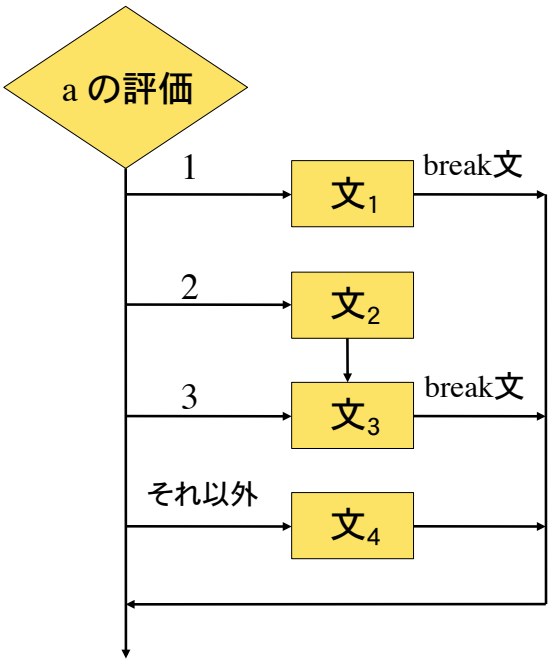
break 文の位置に注意!

break 文を書かないと、次のラベルの行へ処理が移る

default ラベルの break 文は、無くてもよい

switch 文と if 文

前項の switch 文



同じことを if 文で書くと、、、

```
int a;
scanf("%d", &a);

if( a==1 )
  文1
else
  if( a==2 ){
    文2
    文3
  } else
      if( a==3 )
        文3
      else
        文4
```

switch 文の応用

break 文を意図的に省くことで、多様な分岐処理が可能になる

月（int 1 ~ 12 ）を入力して季節を判断する

```
switch(month){
  case 3:
  case 4:
  case 5: printf("春です\n"); break;
  case 6:
  case 7:
  case 8: printf("夏です\n"); break;
  case 9:
  case 10:
  case 11: printf("秋です\n"); break;
  case 12:
  case 1:
  case 2: printf("冬です\n"); break;
  default: printf("エラーです\n");
}
```

ラベルの後に空文（何もしない文）有り

break 文の位置に注意

defaultラベルで、
例外処理も完璧！

条件判断 if 文と switch 文

2 分岐の条件判断: if 文

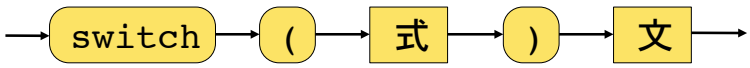
式の真偽で分岐。if 文の入れ子でどのような分岐も可能。

多分岐の条件判断: switch 文

switch 文は、式の値によって分岐先が決まる。
式の値は整数型もしくは文字型に限る。

if 文を switch 文に、switch 文を if 文に書き直すことは可能。しかし、プログラムが読みやすくなるかどうかは別問題。分岐内容によってどちらかを用いる。

switch 文の構文図



文字型

シングルクォーテーション ' で囲った 1 文字を文字定数という

英数文字 A, ?, 1 はプログラム中では 'A' '?' '1' と表記

1 文字を格納する変数の型を文字型といい char で表す

英語で文字を character という

文字型変数の宣言と変数への代入

```
char c;

c = 'A';
```

←― 変数 c に文字 A を代入

文字定数は文字列リテラルとは異なることに注意!

'A' と "A" は違う

ここでの文字とは英数字および記号を指す。
カナや漢字は除く

文字型変数の入出力

文字型変数の入出力(scanf, printf)の変換指定には %c を用いる

```
char a, b;

a = 'A';
scanf("%c", &b);

printf("The 1st character is %c\n", a);
printf("The 2nd character is %c\n", b);
```

文字型変数は英数文字 1 文字を格納する。
上のプログラムの scanf の入力で、1 文字以上の文字を入力しても、最初の 1 文字だけが変数 c に格納される。

例題

次のプログラムの動作を予想せよ

```
char answer;

printf("予習して来ましたか?(Y/N)");
scanf("%c", &answer);
if( answer == 'Y')
  printf("大変結構です\n");
else
  printf("単位落としても知らないよ\n");
```

```
char a;

printf("性別は? M)ale or F)emale ");
scanf("%c", &a);
switch(a){
  case 'M': printf("男性はこちら\n");break;
  case 'F': printf("女性はこちら\n");break;
  default: printf("貴方は何者?\n");
}
```

1 文字の入力（getchar, putchar）

1 文字の入力と出力を行う標準ライブラリ関数として
getchar と putchar がある

```
int c;

c = getchar();
putchar(c);
```

←― キーボードから 1 文字を読み取って整数型変数 c に代入
←― 変数 c を文字として出力

1 文字をなぜ整数型の変数に代入する ???

文字と文字コード

文字（char）は、整数値（int）の文字コードで表される

'0' ゼロ、という文字は、30（16 進数）= 48（10 進数）
'1' 1、という文字は、31（16 進数）= 49（10 進数）

文字と文字コードを対応させる決まりとして、ASCIIコード（JISコード）がある

JIS コード表（16 進数表記）

JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

文字型と整数型の関係

文字型は、整数型の部分集合である。

char　　: 0 ~ 255
int　　: -32768 ~ 32767　(-2147483648 ~ 2147483647)

```
char c;
int i;
```

変数 c は文字型として宣言。格納可能な値の範囲は 0 ~ 255。

1 文字を入力する関数 getchar() は、入力時にエラーが発生したり、特殊な文字が入力されると EOF という値を返す（EOF は –1 という整数値）

そのため、getchar() の返却値を格納する変数は整数型でなくてはならない。

```
c = getchar();
```

← EOF を正しく取り扱えないので不可

```
i = getchar();
```

文字＝文字コード（整数値）

```
int c;

c = 100;
printf("%c %d\n", c, c);

c = 101;
putchar(c);

c = 'A';
putchar(c);
printf("%c %d\n", c, c);

c = getchar();
printf("%c\n", c);

c = getchar();
putchar(c);
```

← 100 (10進数) = 64 (16進数) = 'd'

← 101 (10進数) = 65 (16進数) = 'e'

← 'A' = 41 (16進数) = 65 (10進数)

getchar, putchar

getchar() はキーボードから入力した 1 文字の文字コードを返す関数
特殊文字の扱いのため返却値は整数値

putchar() は、文字コードに対応する 1 文字を出力する関数

```
int c; /* 文字コードを格納する変数は整数型！ */

c = getchar();
printf("文字コード %d に対応する文字は %c です", c, c);
```

```
int c;

c = 100;
putchar(c);
```

文字コード 100 に対応する文字の出力

100 (10 進数) = 64 (16 進数) の文字は 'd'

文字＝文字コード

文字をそのまま取り扱うよりも、文字コードで取り扱ったほうが処理が簡単な場合がある。

JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

例）アルファベット大文字の文字コードの範囲は、41 ~ 5A (16 進数)

A: 16 進数の 41 は、4 * 16 + 1 = 65 (10 進数)

Z: 16 進数の 5A は、5 * 16 + 10 = 90 (10 進数)

```
int c;
c = getchar();
if( c >= 65 && c <= 90 )
  printf("%c は大文字\n", c);
```

```
int c;
c = getchar();
if( c >= 'A' && c <= 'Z' )
  printf("%c は大文字\n", c);
```

日本語の文字コードについて

英数記号は 1 バイト(00~FF の256 通り)で表される。

平仮名、カタカナ、漢字などは 1 バイトでは表すことが出来ない。
通常は 2 バイト(256*256 通りの記述が可能)で表す。

歴史的な経緯により、カナや漢字のコードには複数の体系がある。

JIS コード :インターネット上のメールなど
シフト JIS コード :パソコンで用いられる
EUC コード :Unix で用いられる
Unicode (UTF-8) : 国際化対応の文字コード

たいていの場合、プログラムがコード体系を正しく判断してうまく表示してくれるが、うまくいかず文字化けする場合がある。その時には、コード変換をする必要がある。

10 進数と 16 進数

10 進数:'0' ~ '9' の 10個の記号を用いて表記
16 進数:'0' ~ '9' + 'A', 'B', 'C', 'D', 'E', 'F' の 16個の記号を用いて表記

10進数の 100 を16進数で表記すると、

110 = 6 * 16 + 14 であるので、6E

10進数から16進数への変換方法=16で割り算して商と余り

16進数の 35 を10進数で表記すると、

3 * 16 + 5 = 53 であるので、53

16進数から10進数への変換方法=16でかけ算した合計

printf 整数値の変換指定

%d 値を整数値（10進数）として変換表示

```
int a = 110;
printf("%d\n", a);                    110
```

%x, %X 値を整数値（16進数）として変換表示

```
int a = 110;
printf("%d, %x, %X\n", a, a, a);      100, 6e, 6E
printf("%c\n", a);                    n
```

問題 1

switch 文を用いて、大中小の月を判定するプログラムを作れ
エラー処理も正しく行うこと。

```
% ./a.out
   何月にする? 5
 5 月は 31 日あります
% ./a.out
   何月にする? 9
 9 月は 30 日あります
% ./a.out
   何月にする? 2
 2 月は 28 日、閏年には 29 日あります
% ./a.out
   何月にする? 123
 入力エラーです
%
```

問題 2

次のような受け答えをするプログラムを
1) switch 文、2) if 文を用いて 2 通り作れ

```
% ./a.out
% いらっしゃい！ネタは何にする？
   a) マグロ、b) ヒラメ、c) ウニ、だよ！ a
 マグロは品切れだよ！
% ./a.out
% いらっしゃい！ネタは何にする？
   a) マグロ、b) ヒラメ、c) ウニ、だよ！ b
 ヒラメはカレイと違うんだよ！
% ./a.out
% いらっしゃい！ネタは何にする？
   a) マグロ、b) ヒラメ、c) ウニ、だよ！ 7
 そんなネタないよ！
%
```

この部分はプログラムの出力

問題 3

小文字のアルファベット 1 文字を入力して大文字に変換するプログラム

文字コード表を参考にせよ

```
% ./a.out
 小文字のアルファベット1文字を入力？ a
   a の大文字は A です
% ./a.out
 小文字のアルファベット1文字を入力？ B
 B は小文字のアルファベットではありません
% ./a.out
 小文字のアルファベット1文字を入力？ 9
 9 は小文字のアルファベットではありません
%
```

ヒント

アルファベットの大文字と小文字は
文字コードで 20（16 進数）= 32
（10 進数）の違いがある

小文字のアルファベットの文字コードの
範囲は、61 ~ 7A（16 進数）。それ以外の
コードは小文字のアルファベットではない