

計算物理学II (第6回)

今回の内容

- if文
 - プログラムは上から順番に一文ずつ実行するが、if文によって特定の条件を満たしたときだけ実行することが可能。
 - do文(for文)と組み合わせて使うことで条件が満たされるまで繰り返す、などの制御も可能。
- 詳細は講義資料の文法のまとめを参照

コーディングの環境について

- ホームディレクトリ以外で作業してください。
 - 新しいファイルができたかどうかをlsで探しにくい不是吗。
- emacsは&をつけて実行、書き終わっても終了しない
 - コンパイルが一回で成功することはなかなかありません。
 - 端末とemacs両方を常に使えるようにしておくと効率があがります。
- emacs起動後 **C-x 1**でウィンドウ領域全部を使えます。
- emacsでコードを修正したら保存(**C-x C-s**)
- emacsで新しいファイルを開く場合は**C-x C-f**、その後ファイル名を入力)

コーディングの環境について

- 最近実行したコマンドは”↑”キーを複数回押すとたどれます。
 - コンパイル、実行は通常何度も行いますので毎回gcc/gfortranを入力する必要はありません。
- 端末やemacsは複数起動できます。
 - サンプルを見ながら書く場合はサンプルもemacsで開くとよいです。
- emacs上でTabキーを押すと補完・字下げをしてくれます。
 - Fortran : end do, end programなどの入力時にはendまで入力してTabキー
 - 字下げが正しく行われない場合は手前で文法ミスがある可能性があります。
- 講義資料をpdfで開く場合もうまくサムネイルを小さくする、ページを拡大するなど見やすく調整をしてください。

条件文

- 実行文は上から一行ずつ実行するのが原則だがよく使う例外は2つだけ
- if文：条件を論理式で指定し条件を満たすかどうかで処理内容を変える

Fortran

```
integer :: i  
read (*,*) i  
if ( i<0) i=-i  
print *, "i = ", i
```

C

```
int i;  
scanf("%d", &i);  
if (i<0) i=-i;  
printf("i=%d¥n",i);
```

条件文

Fortran

C

```
if (論理式) 実行文
```

```
if (論理式) then  
  実行文1  
  実行文2
```

```
end if
```

```
if (論理式) then  
  実行文1
```

```
else  
  実行文2
```

```
end if
```

```
if (論理式1) then  
  実行文1
```

```
else if (論理式2) then  
  実行文2
```

```
else  
  実行文3
```

```
end if
```

```
if (論理式) 実行文;
```

```
if (論理式){  
  実行文1;  
  実行文2;  
}
```

```
if (論理式){  
  実行文1;  
}else{  
  実行文2;  
}
```

```
if (論理式1){  
  実行文1;  
}else if (論理式2){  
  実行文2;  
}else{  
  実行文3;  
}
```

論理式が真のときのみ実行文が実行される(実行文が1行の場合)

論理式が真のときのみ実行文1と実行文2が実行される。
実行文が複数行ある場合はこのスタイル

論理式が真の場合は実行文1が実行され、
偽の場合は実行文2が実行される

論理式1が真のときは実行文1を実行
論理式1が偽かつ論理式2が真の時は実行文2を実行
論理式1も論理式2も偽の場合は 実行文3を実行

else ifは複数あってもよい。

論理式

Fortran	Fortran	C	
$a == b$	$a .eq. b$	$a == b$	aがbと等しい時に真 (equal) 等しい時は=一つではなく2つなので注意
$a /= b$	$a .ne. b$	$a != b$	aがbと等しくないとき真(not equal)
$a >= b$	$a .ge. b$	$a >= b$	aがb以上の時に真 (greater equal)
$a > b$	$a .gt. b$	$a > b$	aがbより大きい時に真(greater than)
$a <= b$	$a .le. b$	$a <= b$	aがb以下の時に真(less equal)
$a < b$	$a .lt. b$	$a < b$	aがbより小さい時に真(less than)

等しい時は=一つではなく2つなので注意

Fortran 論理演算子	意味	使用例
.not.	(右)以外	.not. a==b
.and.	かつ	a==b .and. b==c
.or.	もしくは	a==b .or. b==c
.eqv.	論理値が等しい	a<0 .eqv b<0 (aとbの符号が同じ)
.neqv.	論理値が異なる	a<0 .neqv. b<0 (aとbの符号が違う)

C 論理演算子	意味	使用例
!	(右)以外	! a==b
&&	かつ	a==b && b==c
	もしくは	a==b b==c

プログラムの終了

- プログラムは最終行まで到達すれば終了
- 条件文を使うと途中で終了させることも可能
- Fortran: stop文で終了。 stop “文字列” とすると文字列が標準エラー出力に出力
 - プログラムの一番最後にstopと正常終了のメッセージを書く場合もある
- C: return文でmain関数から抜けてプログラム終了。 **異常終了の場合は0でない値を返す。**

Fortran

```
read(*,*) x                                ! xを標準出力から読み込む
if( x<0.0d0) stop "x must be zero or positive" ! xが負であればプログラムを終了
                                                    ! xがゼロか正であれば以下の処理が行われる

print *, "sqrt(x) = ", sqrt(x)
```

C

```
double x;
scanf("%lf", &x);                            // xを標準出力から読み込む
if( x<0.0){
    fprintf(stderr, "x is negative\n");
    return(-1);    // xが負であれば-1を返してプログラムを終了
}
printf("sqrt(x) = %f\n", sqrt(x));            // xがゼロか正であれば以下の処理が行われる
```


条件文の例：クロネッカーのデルタ

Fortran

```
integer :: delta, i, j  
  
read (*,*) i, j  
  
if (i == j) then  
    delta = 1  
else  
    delta = 0  
end if  
  
print *, delta
```

C

```
int delta, i, j;  
  
scanf("%d %d", &i, &j);  
  
if (i == j){  
    delta = 1;  
}else{  
    delta = 0;  
}  
  
printf("delta=%d\n", delta);
```

Fortran

```
integer :: delta, i, j  
  
read(*,*) i,j  
  
delta = 0  
if(i==j) delta = 1  
  
print *, delta
```

C

```
int delta, i, j;  
  
scanf("%d %d", &i, &j);  
  
delta = 0;  
if(i==j) delta = 1;  
  
printf("delta = %d\n", delta);
```

doループとif文の組み合わせ

無限ループ

```
do
!ここでいろいろ計算
if ( a<0 ) exit
x=x+0.10d0
end do
```

```
for(x=0.1; ; x+=0.1){
// ここでいろいろ計算
a = ....;
if ( a<0 ) break;
}
```

exit/break文はループから抜け、ループ文の直後に移動

条件が満たされない場合は計算が終わらないため実行してみて終わらない場合は**強制終了(C-c)する**

cycle(Fortran), continue(C) : ループの先頭まで戻り、カウンタを次の値に進める

```
do i = 1, n
```

```
  a(i) = ....ここで何かを計算する
```

```
  if ( a(i)>0 ) cycle
```

```
  !a(i)が正なら戻ってカウンタを一つ回す
```

```
  ! a(i)が0か負の場合だけこの領域で処理が行われる
```

```
end do
```

```
for( i = 0;i<n;i++){
```

```
  a[i] = ....; // ここで何かを計算する
```

```
  if (a[i]>0) continue; //a[i]>0なら戻ってカウンタを一つ回す
```

```
  /* a[i]が0か負の場合だけこの領域で処理が行われる */
```

```
}
```

演習

- 自分で書けそうなら資料を見て、難しそうならサンプルのプログラムを写して書いてコンパイル、実行してください。
- 条件文は第6回の(8)-(11)
- 第1回のレポート締切は11月17日(金)です。
- 質問がある方はTeamsのチャットで質問してください。