



「・ω・」



C言語講座

配列・演算子・分岐・ループ

neko[0]



配列

- データをまとめて管理できる
- データ型 配列名[要素の数];

例: `int a[5];` 数を入れる
`char s[5];` 文字を入れる

neko[1]

[0]から始まる

[]の中には要素の番号が入る

char型の末尾にはNULL(ヌル)
が入り、
¥0 で表します



多次元配列

- データ型 配列名 [横の要素数][縦の要素数];

例: `int a[2][3];`

二次元配列

三次元以上の配列はあまり使わない

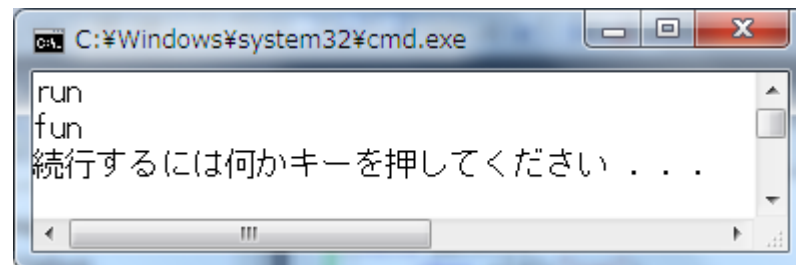


例題

下記ソースコードに一行加え、rだけを変更してrunをfunにしてください

(※s[4]="fun";を加えるのは不可)

```
#include<stdio.h>
void main(){
    char s[4]="run";
    printf("%s\n", s); //runと表示
    s[0]='f; //rをfに変更
    printf("%s\n",s); //funと表示
}
```



```
C:\Windows\system32\cmd.exe
run
fun
続行するには何かキーを押してください . . .
```



演算子

- 比較演算子

== (等しい)

!= (等しくない)

<, <=, >=, > (大小関係)

- 論理演算子

! (NOT:否定)

&& (AND:かつ)

|| (OR :または)

= ではなく
== と記述する

代入と判断されてしまう



演算子

正しい演算は真(1:True)

異なる演算は偽(0:False) を示す(返す)

例:(2==3)&&(4<5) は 0 となる



0 && 1



例題

1 (True) と 0 (False) のどちらを示す (返す) か

- $2 < 3$ 1
- $4 == 5$ 0
- $(2 < 3) \ || \ (4 == 5)$ 1



ループ

- 同様の処理を繰り返す

繰り返す前に
条件を調べる

- while(条件){
.....(略)
}

条件を満たす間 繰り返す

- do{

「;」あるよ!

.....(略)

}while(条件);

繰り返してから
条件を調べる



例題

```
#include<stdio.h>
void main(){
    int i=10;
    while(i<10){

        printf(“%d”,i);
        i++;
    }
}
```

表示されない

それぞれの実行結果はどうなるでしょうか

```
#include<stdio.h>
void main(){
    int i=10;
    do{

        printf(“%d”,i);
        i++;
    }while(i<10);
}
```

10 が表示される

ループ

- for(初期値;条件;カウンタ){
.....(略)
}

「;」の位置に注意

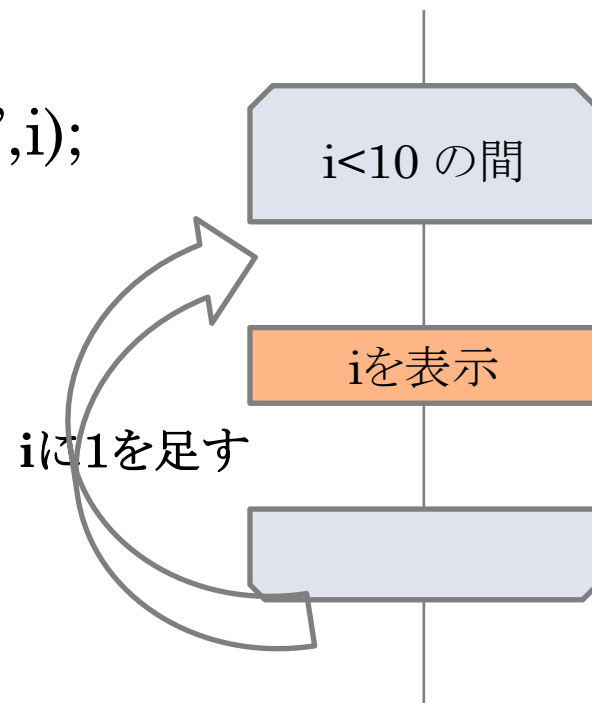
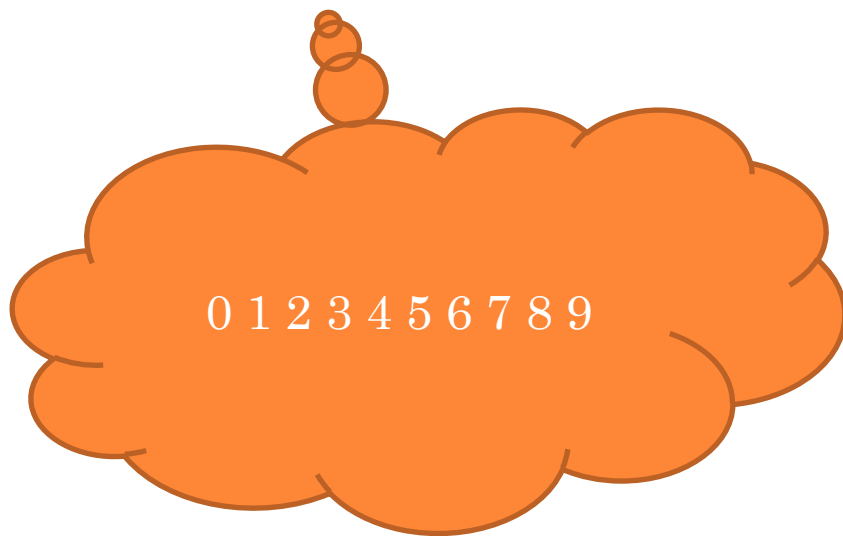
for文...〇回処理を繰り返す
while文...～の間処理を繰り返す



ループ

例: `for(i=0; i<10; i++){`

```
printf(“%d”,i);  
}
```



例題

while(条件) 、 for(初期値;条件;カウンタ)

において、無限ループにするにはそれぞれどうするか

答え

while(1){(略)

for(;;){(略)

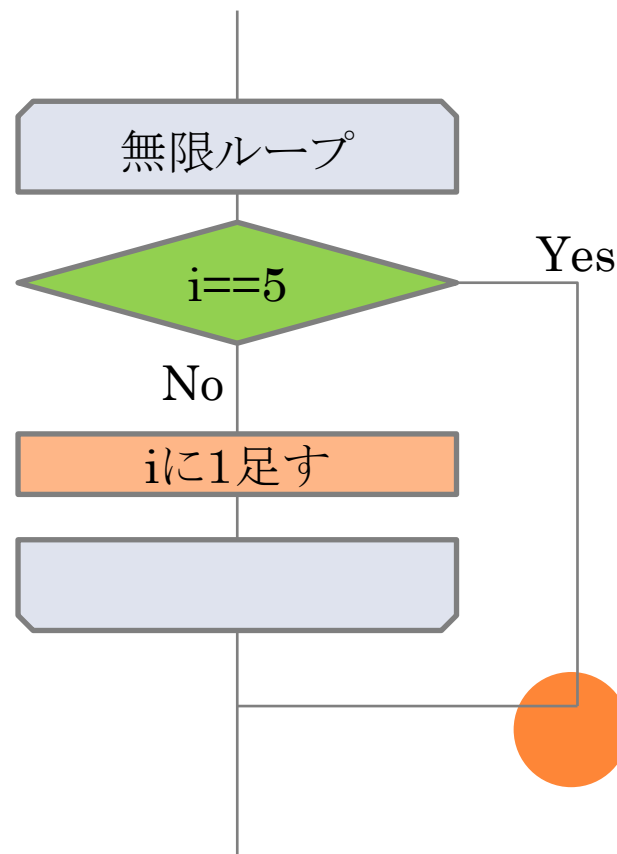


while(0)なら常に条件を満たさない
=実行されない

BREAK

- ループ処理を抜け出せる
- 1つにつき1個のループから抜ける

```
例: while(1){  
    if(i==5){break;}  
    i++;  
}
```

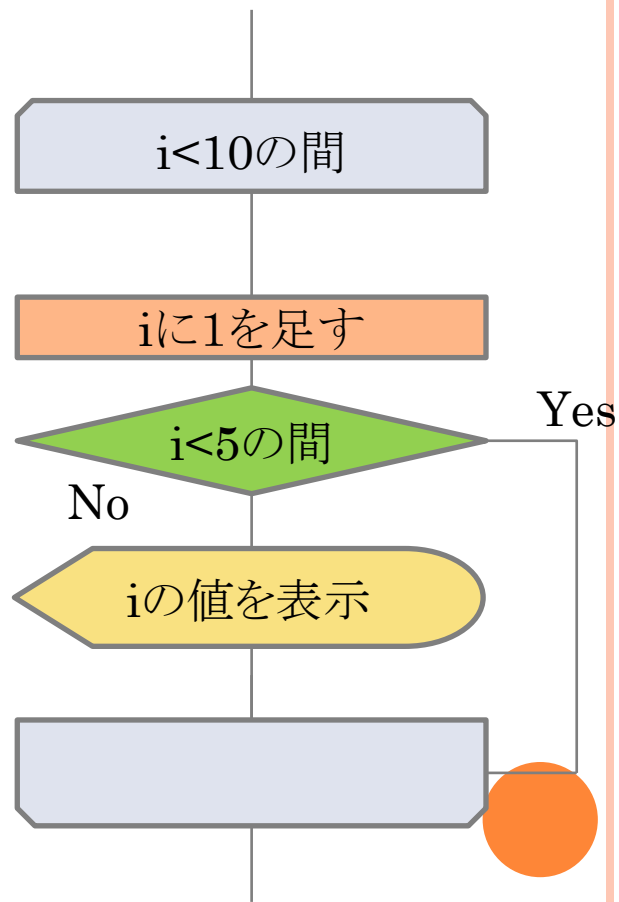


CONTINUE

- breakと違い、完全にループを抜けるのではなく一時的にループを抜ける

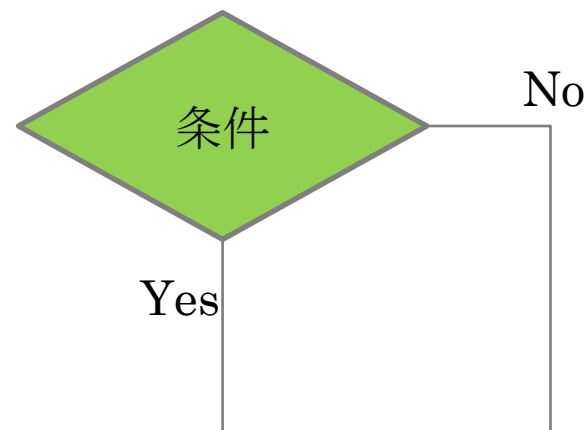
```
例: while ( i < 10 ) {  
    i ++;  
    if ( i < 5 ) { continue;  
    printf(“%d”, i);  
    }  
}
```

5 6 7 8 9 10



分岐 (IF)

- 条件によって処理を変える
- `if(条件){`
.....(略)
}
- `else if(条件){`
.....(略)



2つ目以降の条件は
`else if(条件)`
で追加



score に得点を代
入

2つ以上の演算は
&& または || で行
う

```
例: if(score >= 80){  
    printf("得点は80点以上です¥n");  
}  
else if(30 < score && score < 80){  
    printf("得点は30~80点です¥n");  
}  
else{  
    printf("得点は30点以下です¥n");  
}
```



分岐(SWITCH)

- 条件によって処理を変える

- `switch(条件){`

- `case 定数1:`

- `.....(略)`

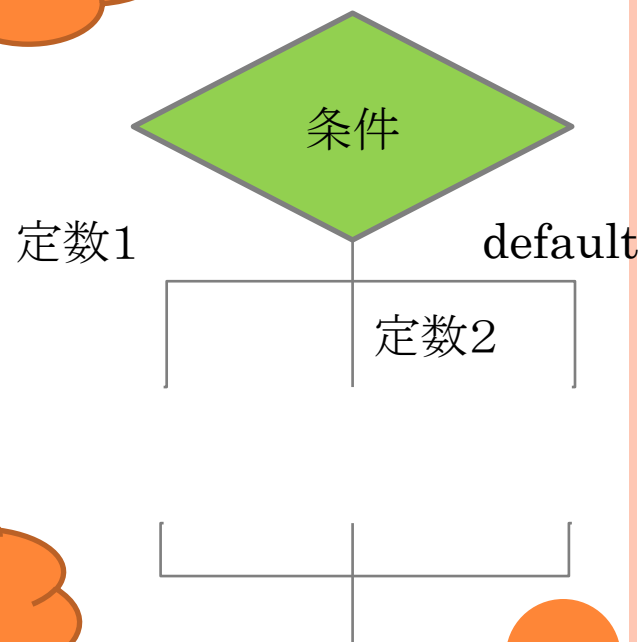
- `break;`

- `case 定数2:`

- `.....(略)`

- `break;`

- `}`



分岐(SWITCH)

```
例: switch(score){  
    case 100:  
        printf("得点は100点です¥n");  
        break;  
    case 50:  
        printf("得点は50点です¥n");  
        break;  
    case 0:  
        printf("得点は0点です¥n");  
        break;  
    default:  
        printf("それ以外の得点です¥n");  
        break;  
}
```



default は省略可



総合問題

評価A+,A,B,C,D,E のいずれかを入力すると、その成績評価をとるために必要な基準得点の割合が表示されるプログラムを作成してください

ただし評価の入力は5回までとします

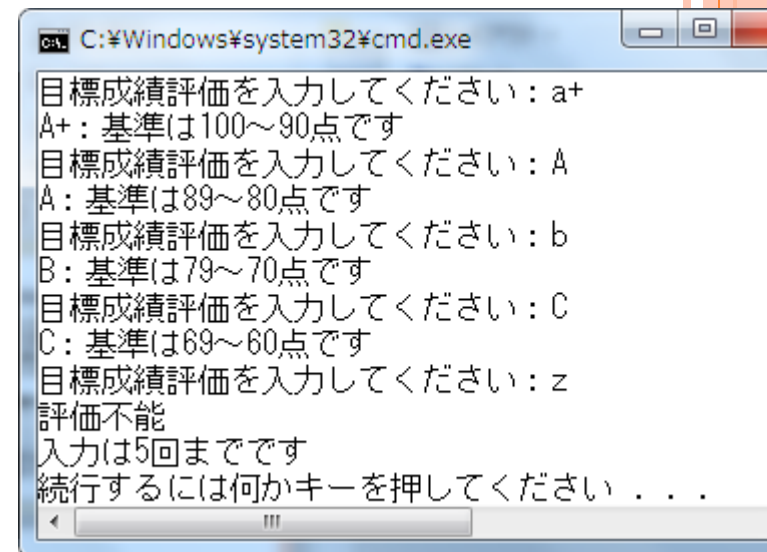
評価の入力には `char s[5]` を使用してください

○ 評価基準

A+...100～90点 /

A...89～80点 / B...79～70点/

C...69～60点 / D...59～0点



```
C:\Windows\system32\cmd.exe
目標成績評価を入力してください : a+
A+ : 基準は100～90点です
目標成績評価を入力してください : A
A : 基準は89～80点です
目標成績評価を入力してください : b
B : 基準は79～70点です
目標成績評価を入力してください : C
C : 基準は69～60点です
目標成績評価を入力してください : z
評価不能
入力は5回までです
続行するには何かキーを押してください . . .
```

ヒント

- if文でelse if
- s[0]が1文字目、s[1]が2文字目
- 判断には最後の文字'¥0'も利用
→評価A は s[0]が'A'、s[1]が'¥0'
- ANDがORより優先なのでOR優先にしたい時は()をつける
- 5回入力を繰り返す(ループ)



解答例

```
#include<stdio.h>
void main(){
    int i;
    char s[5];

    for(i=0;i<5;i++){
        printf("目標成績評価を入力してください:");
        scanf("%s",s);

        if( (s[0]=='A' || s[0]=='a') && s[1]=='+' )
            printf("A+: 基準は100～90点です¥n");
        else if( (s[0]=='A' || s[0]=='a') && s[1]=='¥0' )
            printf("A: 基準は89～80点です¥n");
```



解答例

```
else if( (s[0]=='B' || s[0]=='b') && s[1]=='¥0' )
    printf("B:基準は79～70点です¥n");
else if( (s[0]=='C' || s[0]=='c') && s[1]=='¥0' )
    printf("C:基準は69～60点です¥n");
else if( (s[0]=='D' || s[0]=='d') && s[1]=='¥0' )
    printf("D:基準は59～0点です¥n");
else
    printf("評価不能¥n");
}
printf("入力は5回までです¥n");
}
```





お疲れ様でした！