知能情報工学演習I
第12回（後半第6回）

岩村雅一
masa@cs.osakafu-u.ac.jp

前々回の課題で実際にあった間違い

- 課題1：偶数か奇数かチェックするプログラム

```
int a;
printf("自然数を入力して下さい。¥n");
scanf("%d",&a);

if(a/2==(float)a/2){
  printf("偶数です。¥n");
}else{
  printf("奇数です。¥n");
}
```

10人

前々回の課題で実際にあった間違い

- 課題1 : 偶数か奇数かチェックするプログラム
  - if(a/2==(float)a/2)の挙動
  - a=2のとき
    - a/2 = 1　　(float)a/2 = 1.000000
  - a=3のとき
    - a/3 = 1　　(float)a/3 = 1.500000

(float)a/2 = 0.999999になる可能性がある

前々回の課題で実際にあった間違い

- 課題3：キーボードから5つの整数を入力し、5つの値の合計、平均を小数で求める。
  - 配列の大きさを間違えている（2人）

```
int a[4];
```

  - 結果は小数だけど、整数で演算している（10人）

```
int a,b,c,d,e,sum;
float ave;
/* 値の入力 */
sum = a+b+c+d+e;
ave = sum/5;
```

float　int　int

演算のルール

int = int / int

前々回の課題で実際にあった間違い

- 課題４：キーボードから入力した整数が素数かどうかを判定する 。
  - １を素数とした（６人）
  - １を対象外とした（２人）

前回の課題

- 課題1
  - キーボードから5つの整数を入力し、小さい順に並べ替えなさい。変数aとbの値を交換するときには以下の方法がよく使われる。

```
int a, b, tmp;
tmp=a;
a=b;
b=tmp;
```

  - ヒント：for文を二重に使うとよい。

前回の課題：課題1

```
#include <stdio.h>

int main(void){
  int a[5],tmp,i,j;

  for(i=0;i<5;i++){
    printf("%d番目の数字: ",i+1);
    scanf("%d",&a[i]);
  }

  for(i=0;i<5;i++){
    for(j=i+1;j<5;j++){
      if(a[i]>a[j]){
        tmp=a[i];
        a[i]=a[j];
        a[j]=tmp;
      }
    }
  }

  printf("小さい順に並べた結果： ");
  for(i=0;i<4;i++){
    printf("%d,",a[i]);
  }
  printf("%d¥n",a[4]);
  return(0);
}
```

課題１で実際にあった間違い

- 配列の範囲外を使用（２人）
- 宣言する配列の大きさを間違えている（7人）
  - int a[4];
- 結果が出力されたり、されなかったり（１人）
  - 結果の出力がfor文の中にある
- 表示がおかしい（２人）
  - 1,2,3,4,5と入力すると、「12345」

課題1で実際にあった間違い(じゃないけど...)

- 間違いじゃないけど、配列の添え字を1から使っている(2人)
- 間違いじゃないけど、必要以上に配列を宣言している(6人くらい)
  - int a[6];
  - int a[44];

前回の課題

- 課題2
  - キーボードから3桁の自然数を入力したとき、1の位、10の位、100の位に同じ数字がちょうど2つあるかどうかを調べよ。

```
#include<stdio.h>

int main(void){
  int a;

  printf("数字を入力してください。¥n");
  scanf("%d",&a);

  if(a%2==1){
    printf("数字は奇数です。¥n");
  }else{
    printf("数字は偶数です。¥n");
  }
  return 0;
}
```

前回の課題：課題2

```
#include<stdio.h>

int main(void){
  int a,A,B,C;

  printf("3けたの自然数を入力して下さい
     ¥n1の位、10の位、100の位に同じ数字
     がちょうど2つあれば真,そうでなければ
     偽¥n");

  scanf("%d",&a);
  A=a/100;
  B=(a%100)/10;
  C=(a%10);

if (A==B&&B==C){
     printf("偽¥n");
  } else if (A==B||B==C||C==A){
     printf("真¥n");
  } else{
     printf("偽¥n");
  }

return(0);
}
```

課題２で実際にあった間違い

- 判定できない（３人；おしいのを含む）
- 同じ数字が3つか2つか判定しない（２人）
- 同じ数字が3つあっても2つ判定する（１人）

前回の課題

- 課題3
  - キーボードから自然数を入力してもらい、その数を素因数分解しなさい。

前回の課題：課題3

```
#include<stdio.h>
int main(void){

 int m,n;
 int c = ' ';
 printf("整数を入力してください。");
 scanf("%d",&n);
 printf("%d=",n);

 if (n==1) {
   printf("1¥n");
   return(0);
 }
```

```
for (m=2; n!=1 ;m++){
    while(n%m == 0) {
     n = n/m;
     printf("%c%d",c,m);
    c = '*';
      }
 }
 printf("¥n");
 return(0);
}
```

課題3で実際にあった間違い

- 素因数に1が入っている（４人）
- １を素因数分解してくれない（３人）
- 最後の素因数が出ない（１人）
- 「floating point exception」が出る（１人）

後半の予定

7. 6月1日　プログラミング環境（テキスト1,2章）
8. 6月8日　変数とデータ型（3章）、演算子（4章）
9. 6月15日　コンソール入出力（6章）、配列（3章）
10. 6月22日　制御文1（テキスト5章）
11. 6月29日　制御文2（テキスト5章）
12. 7月13日　関数（テキスト7章）
13. 7月20日　配列（3章）、応用プログラム

本日のメニュー

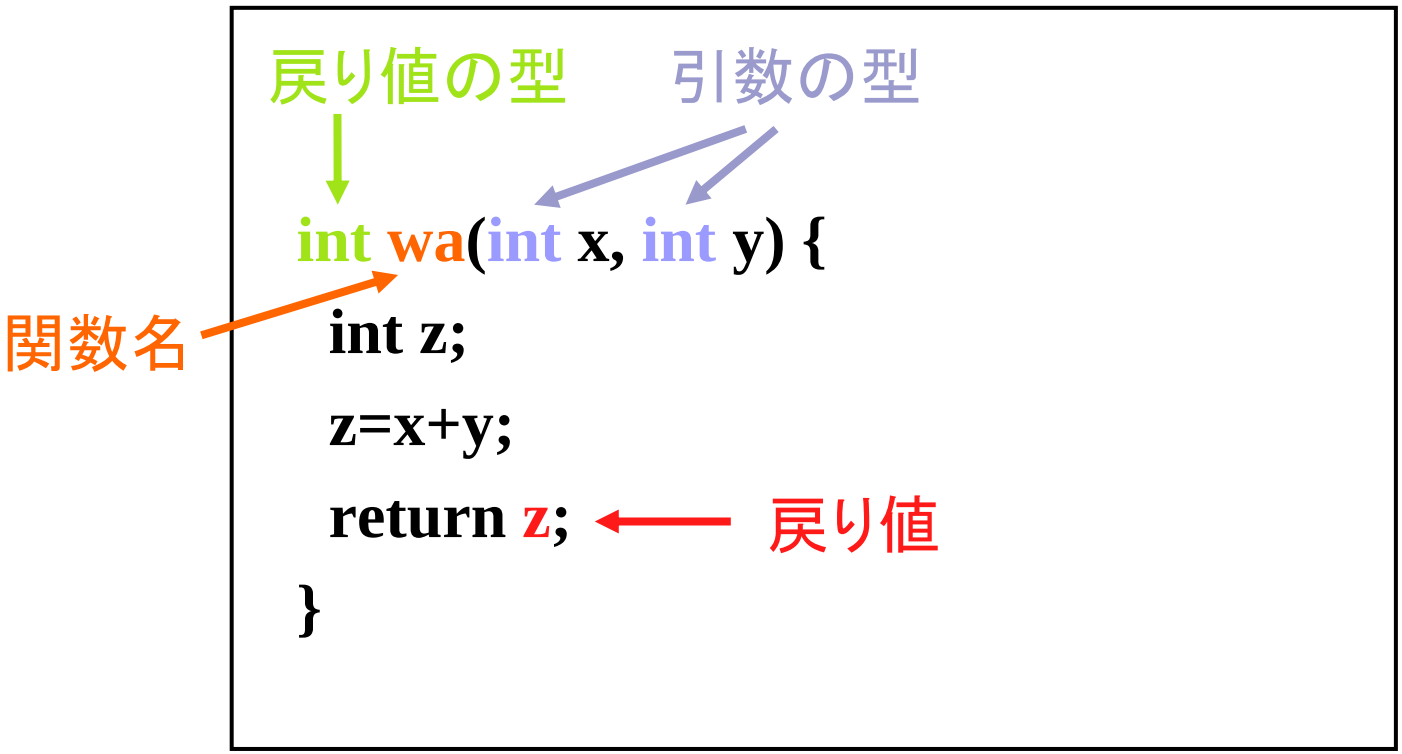
- 関数
  - 関数とは何か
  - 引数や戻り値が無い関数
  - return文
  - 関数の再帰的呼び出し
  - ローカル変数とグローバル変数
  - プロトタイプ宣言

関数とは

- 引数を取り、処理の後、戻り値を返すもの
  - 数学の関数(”y=f(x)”)のようなもの
  - printf()やscanf()は関数
  - main()も関数

関数の例（定義）

- 関数waは、2つの引数xとyを元にzの値を計算し、戻り値として返す



サンプルプログラム

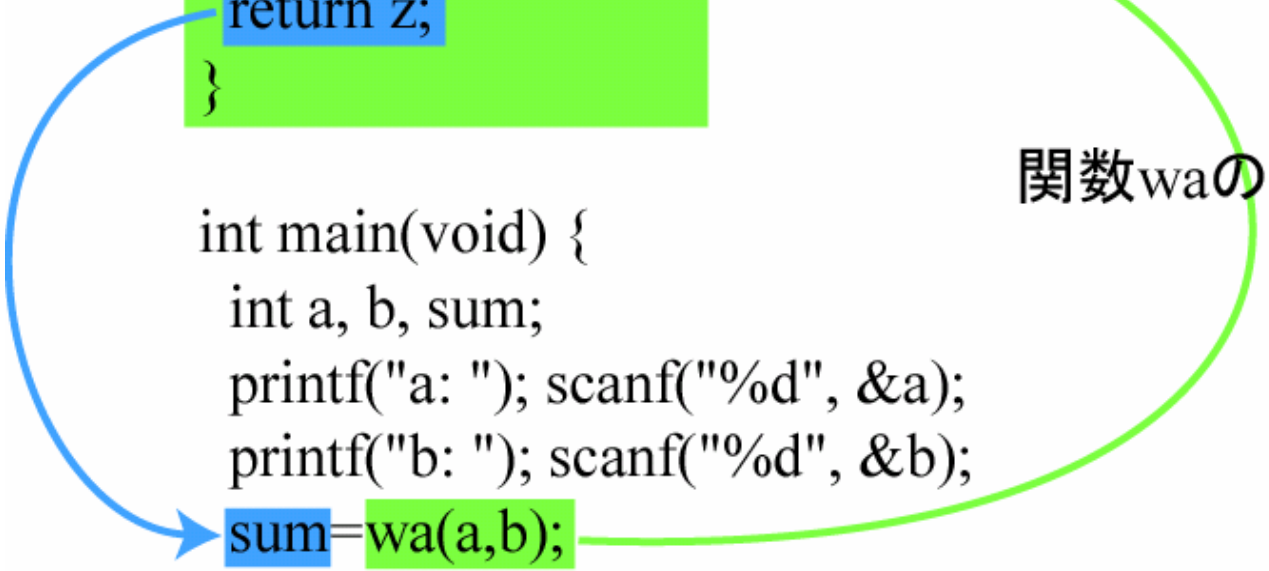
```
#include <stdio.h>

int wa(int x, int y) {
  int z;
  z=x+y;
  return z;
}

int main(void) {
  int a, b, sum;
  printf("a: "); scanf("%d", &a);
  printf("b: "); scanf("%d", &b);
  sum=wa(a,b);
  printf("a + b = %d\n", sum);

  return 0;
}
```

サンプルプログラム

```
#include <stdio.h>

int wa(int x, int y) {
  int z;
  z=x+y;
  return z;
}

int main(void) {
  int a, b, sum;
  printf("a: "); scanf("%d", &a);
  printf("b: "); scanf("%d", &b);
  sum=wa(a,b);
  printf("a + b = %d\n", sum);

  return 0;
}
```

```
int x=a;
int y=b;
```

関数waの呼び出し



引数や戻り値が無い関数

- 引数が無い関数

```
int func(void) {
  …
}
```

- 戻り値が無い関数

```
void func(int a) {
  …
}
```

return文の役割

- return文があると、関数の処理を打ち切って呼び出し元に戻る

```
int waru(int x, int y) {
  int z;
  if (y==0) {
  return -999;
}
  z=x/y;
  return z;
}
```

yが0ならば-999を返して関数の計算は終了

階乗を計算する関数

- 階乗を計算する関数を考えてみよう
  - 3! = 1 * 2 * 3 = 6
  - 0! = 1

```
#include <stdio.h>

int fact(int x) {
  int fact = 1;
  …
  return(fact);
}
```

```
int main (void){
  int n;

  printf("n: "); scanf("%d", &n);

  printf("fact(%d) = %d¥n",  n,
    fact(n));

  return(0);
}
```

階乗を計算する関数

```
#include <stdio.h>

/* 階乗を計算する関数 */
int fact(int x) {
  int i, fact = 1;
  for(i = 2; i <= x; i++) {
    fact *= i;
  }
  return(fact);
}
```

fact = 1 * 2 * 3 * 4 *…

0!や1!もok

```
int main (void){
  int n;

  printf("n: "); scanf("%d", &n);

  printf("fact(%d) = %d¥n",  n,
    fact(n));

  return(0);
}
```

関数の再帰的呼び出し

```
#include <stdio.h>

/* 階乗を計算する関数 */
int fact(int x) {
  if (x==1 || x==0) {
    return(1);
  } else {
    return(x*fact(x-1));
  }
}
```

```
int main (void){
  int n;

  printf("n: "); scanf("%d", &n);

  printf("fact(%d) = %d¥n",  n,
    fact(n));

  return(0);
}
```

関数の再帰的呼び出し

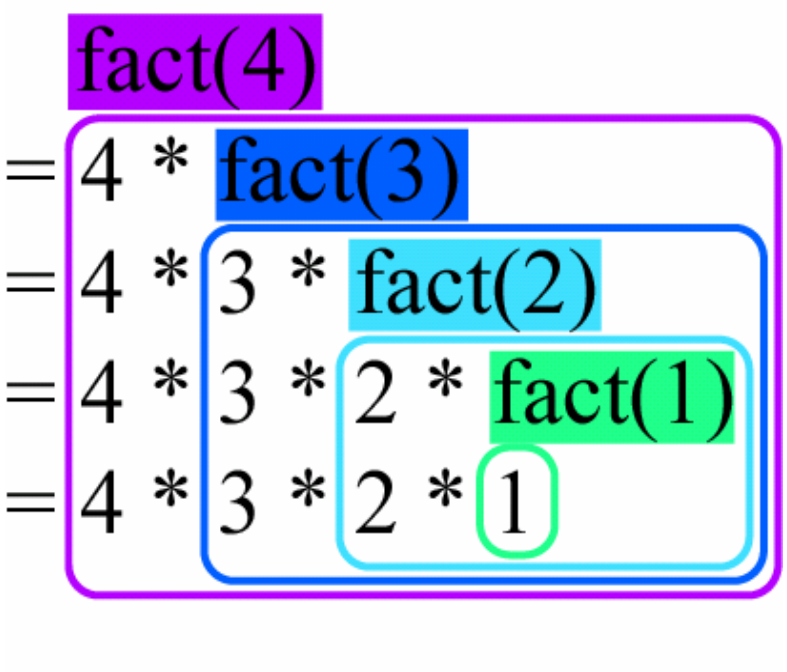
```
#include <stdio.h>

/* 階乗を計算する関数 */
int fact(int x) {
  if (x==1 || x==0) {
    return(1);
  } else {
    return(x*fact(x-1));
  }
}
```

```
/* 階乗を計算する関数 */
int fact(int 1) {
  if (x==1 || x==0) {
    return(1);
  } else {
    return(1*fact(1-1));
  }
}
```

fact(1)
=
1

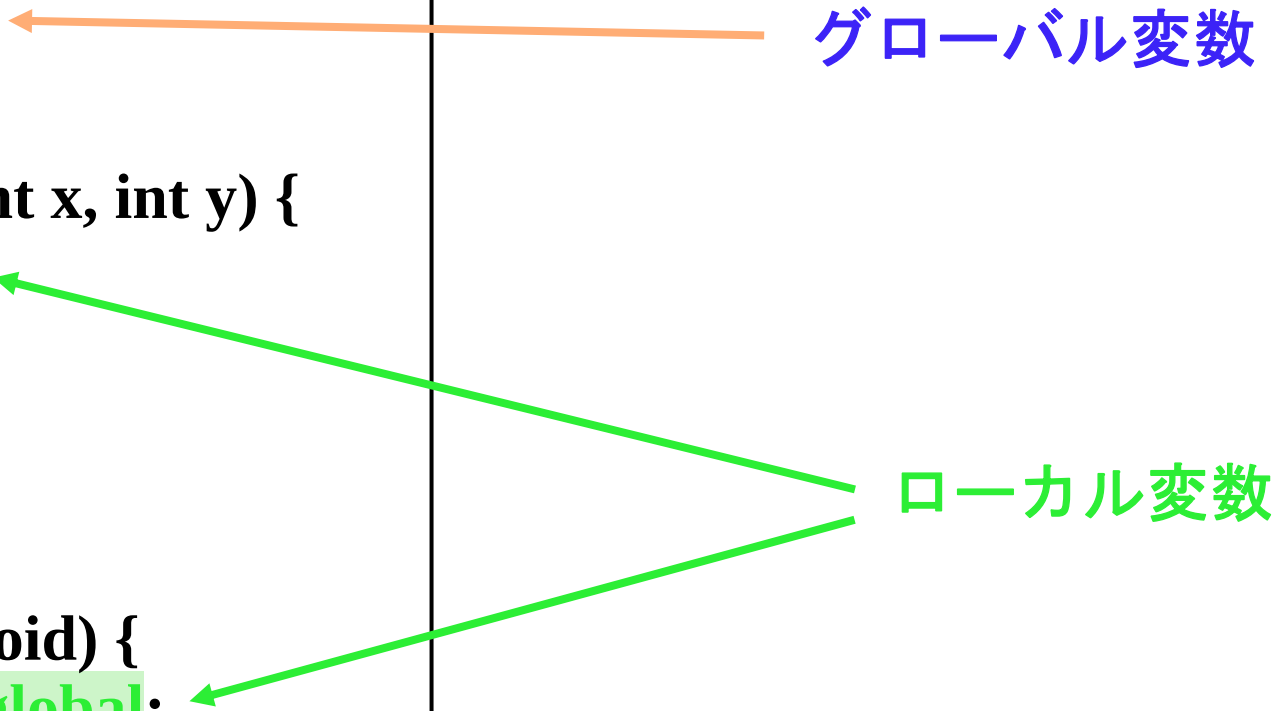
関数の再帰的呼び出し

```
#include <stdio.h>

/* 階乗を計算する関数 */
int fact(int x) {
  if (x==1 || x==0) {
    return(1);
  } else {
    return(x*fact(x-1));
  }
}
```



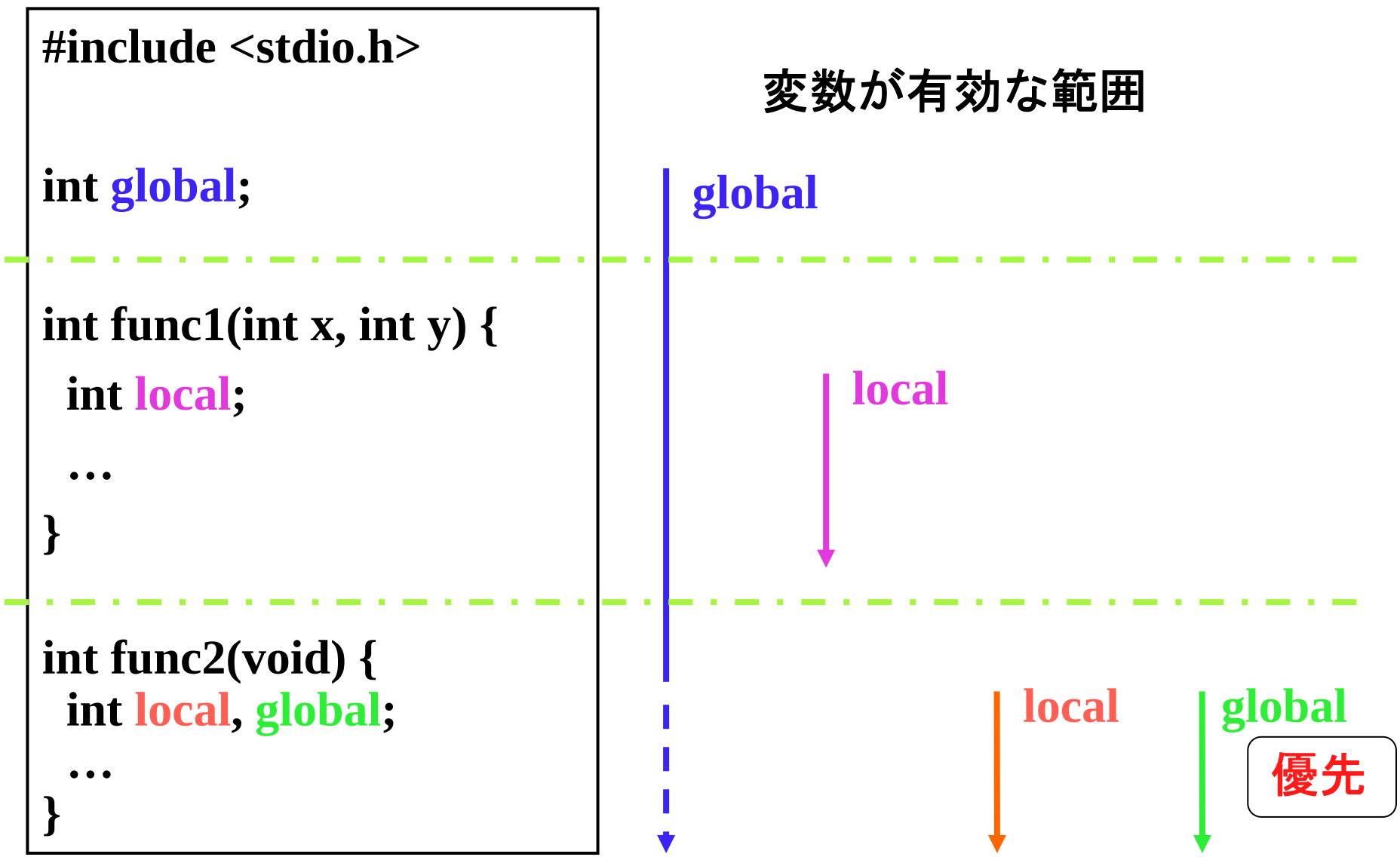
ローカル変数とグローバル変数

```
#include <stdio.h>

int global;

int func1(int x, int y) {
  int local;
  …
}

int func2(void) {
  int local, global;
  …
}
```



ローカル変数とグローバル変数

```
#include <stdio.h>

int global;

int func1(int x, int y) {
  int local;
  …
}

int func2(void) {
  int local, global;
  …
}
```



プロトタイプ宣言

- 関数の「定義」を先にできない場合は「宣言」だけを先に書く

プロトタイプ宣言

移動

```
#include <stdio.h>


int main(void) {
  int a, b, sum;
  printf("a: "); scanf("%d", &a);
  printf("b: "); scanf("%d", &b);
  sum=wa(a,b); ???
  printf("a + b = %d\n", sum);

  return 0;
}

int wa(int x, int y) {
  int z;
  z=x+y;
  return z;
}
```

プロトタイプ宣言

```
#include <stdio.h>

int wa(int x, int y);

int main(void) {
  int a, b, sum;
  printf("a: "); scanf("%d", &a);
  printf("b: "); scanf("%d", &b);
  sum=wa(a,b);
  printf("a + b = %d\n", sum);

  return 0;
}

int wa(int x, int y) {
  int z;
  z=x+y;
  return z;
}
```