

Ark Oak企画

# C言語入門

第5回



# 前回の復習



おーく君



# ファイル操作って？

.txt や .csv のようなファイルの入出力操作

これまで（メモリ操作）

printf() … 画面に文字の出力  
scanf() … キーボードから入力

これから（ファイル操作）

fprintf() … ファイルへ出力  
fscanf() … ファイルから入力

ファイル操作ができれば、一気に活用の幅が広がる！



# ファイル操作の流れ

1. ファイルポインタの宣言

```
FILE *fp;
```

2. ファイルを開く

```
fp = fopen("hoge.txt", "w");
```

3. ファイルを読み書きする

```
fprintf(fp, "hello");
```

4. ファイルを閉じる

```
fclose(fp);
```

おまじないタイム



# ファイルを開くモード

```
fp = fopen("hoge.txt" , "w");
```

ファイル名 モード

| モード | 動作                | ファイルが無いとき |
|-----|-------------------|-----------|
| "w" | 書き込みモードでファイルを開く   | 新規作成      |
| "r" | 読み込みモードでファイルを開く   | エラー       |
| "a" | 追加書き込みモードでファイルを開く | 新規作成      |

"w"モードで既にファイルがあるときは、前の内容は破棄されます





# ファイルに書き込んでみよう

```
#include <stdio.h>

int main(void)
{
    //ファイルポインタの宣言
    FILE *fp;

    //ファイルを開く
    fp = fopen("hello.txt", "w");

    //ファイルに書き込む
    fprintf(fp, "Hello World");

    //ファイルを閉じる
    fclose(fp);
    return 0;
}
```

やってみよう



## 宿題

極座標  $r = \theta \sin(\cos \theta)$  ( $-2\pi \leq \theta \leq 2\pi$ ) のグラフを描いてみよう

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159265358979

int main(void)
{
    FILE *fp;
    fp= fopen("graph.csv", "w");

    //ファイルに書き込む
    double r, x, y, t;

    fclose(fp);
    return 0;
}
```

極座標から直交座標への変換は覚えてるかな？



## 宿題

極座標  $r = \theta \sin(\cos \theta)$  ( $-2\pi \leq \theta \leq 2\pi$ ) のグラフを描いてみよう

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159265358979

int main(void)
{
    FILE *fp;
    fp= fopen("graph.csv", "w");

    //ファイルに書き込む
    double r, x, y, t;

    fclose(fp);
    return 0;
}
```

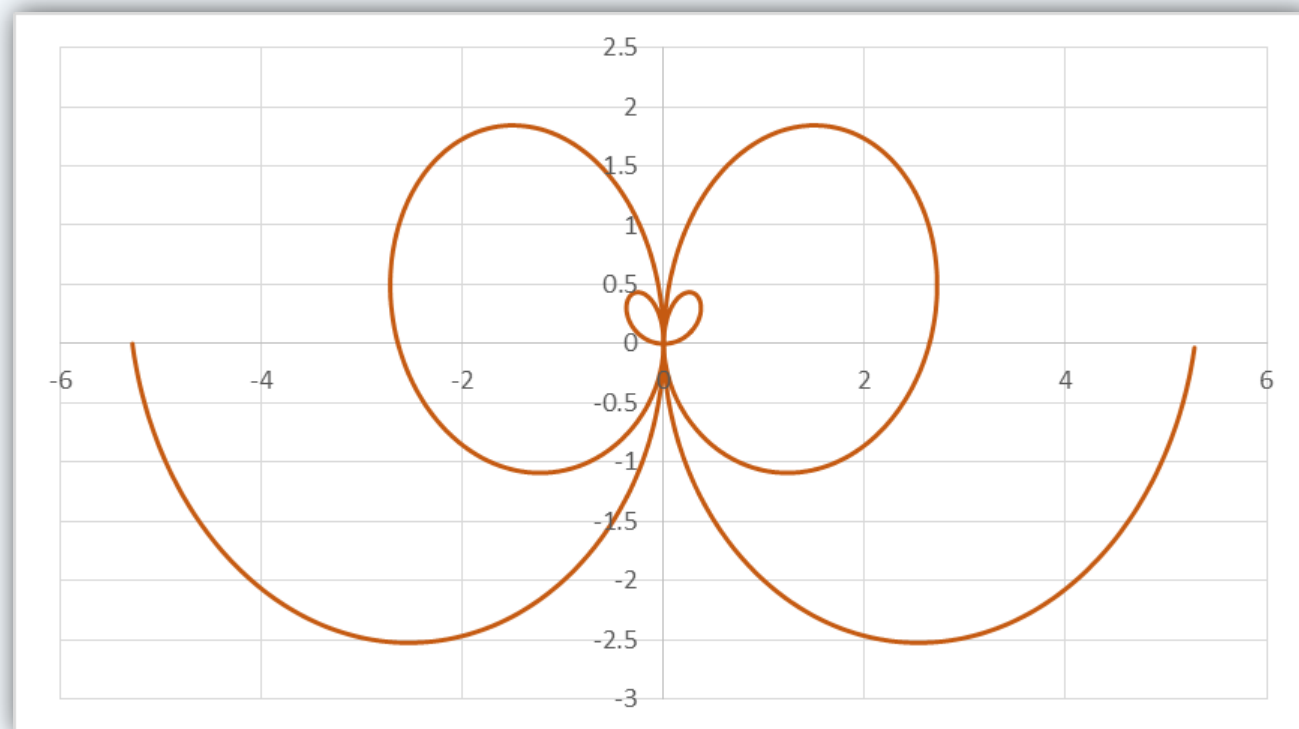
極座標から直交座標への変換は覚えてるかな？





## 宿題

極座標  $r = \theta \sin(\cos \theta)$  ( $-2\pi \leq \theta \leq 2\pi$ ) のグラフを描いてみよう



謎グラフ



# おまけ

$x^2 + (y - \sqrt[3]{x^2})^2 = 1$  のグラフを描いてみよう

変形させると、

$$y = \pm\sqrt{1 - x^2} + \sqrt[3]{x^2}$$

これを描こう

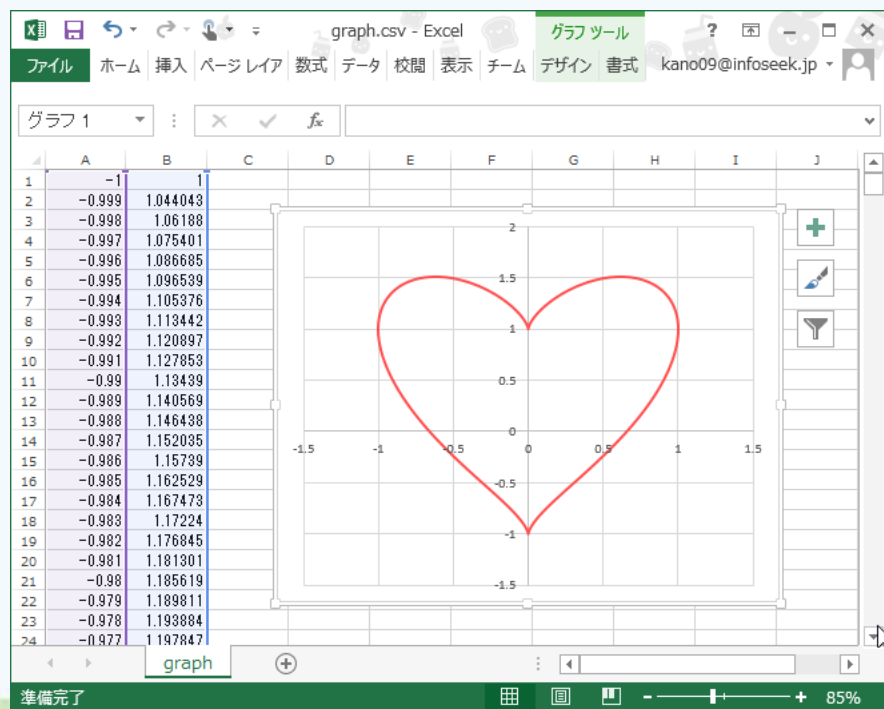
The Love Formula (愛の方程式)  
として知られる曲線だよ



# 復習

## おまけ

$$y = \pm\sqrt{1-x^2} + \sqrt[3]{x^2}$$



```
#include <stdio.h>
#include <math.h>
#define PI 3.1415926535
```

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

```
    FILE *file;
    file = fopen("graph.csv", "w");
```

```
    double x, y, t;
    for(x = -1; x <= 1; x += 0.001){
        y = sqrt(1-x*x) + pow(x*x, 1.0/3.0);
        fprintf(file, "%lf, %lf\n", x, y);
```

```
    }
    for(x = 1; x >= -1; x -= 0.001){
        y = -sqrt(1-x*x) + pow(x*x, 1.0/3.0);
        fprintf(file, "%lf, %lf\n", x, y);
```

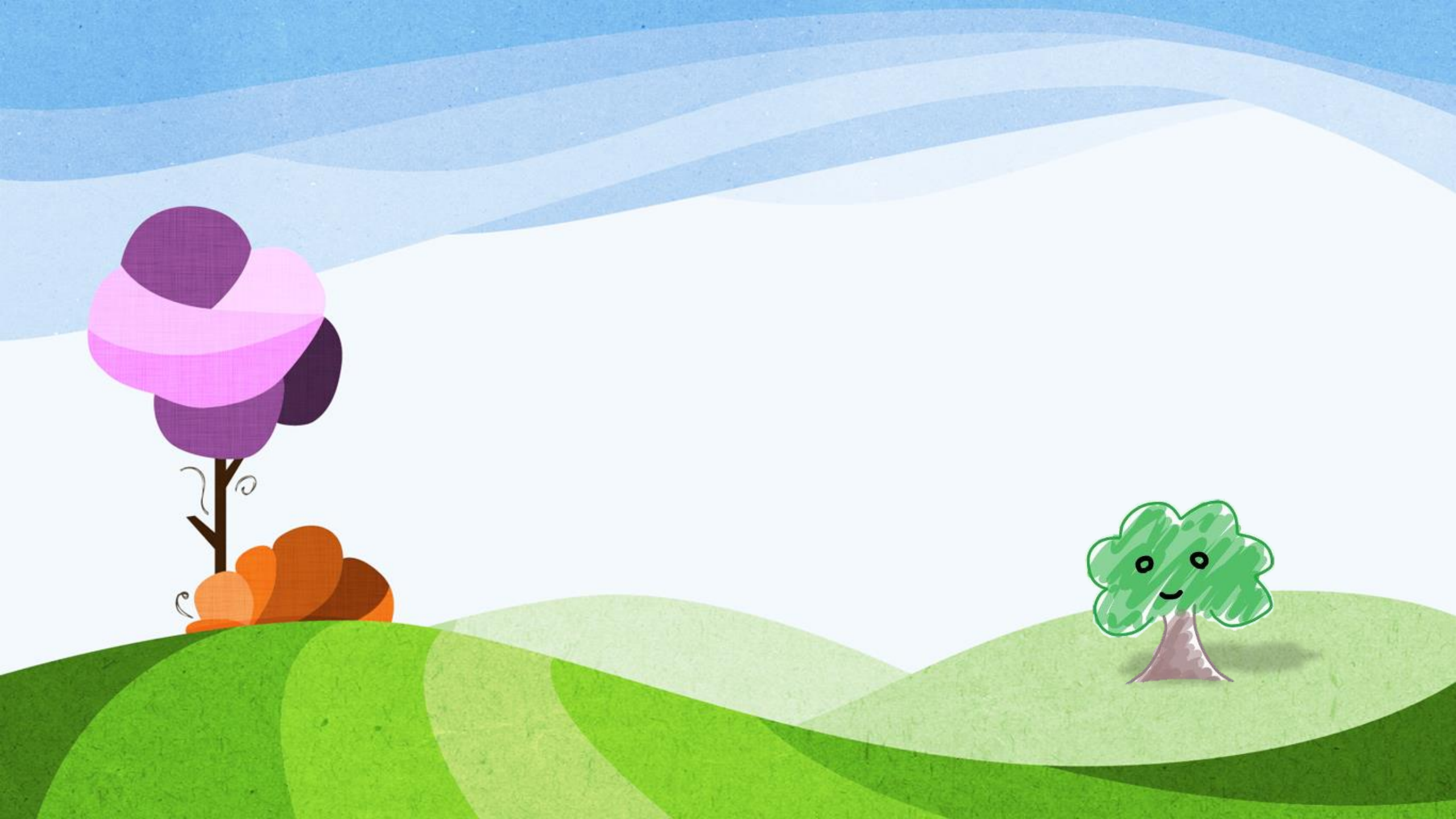
```
    }

    fclose(file);
    return 0;
```

```
}
```

...



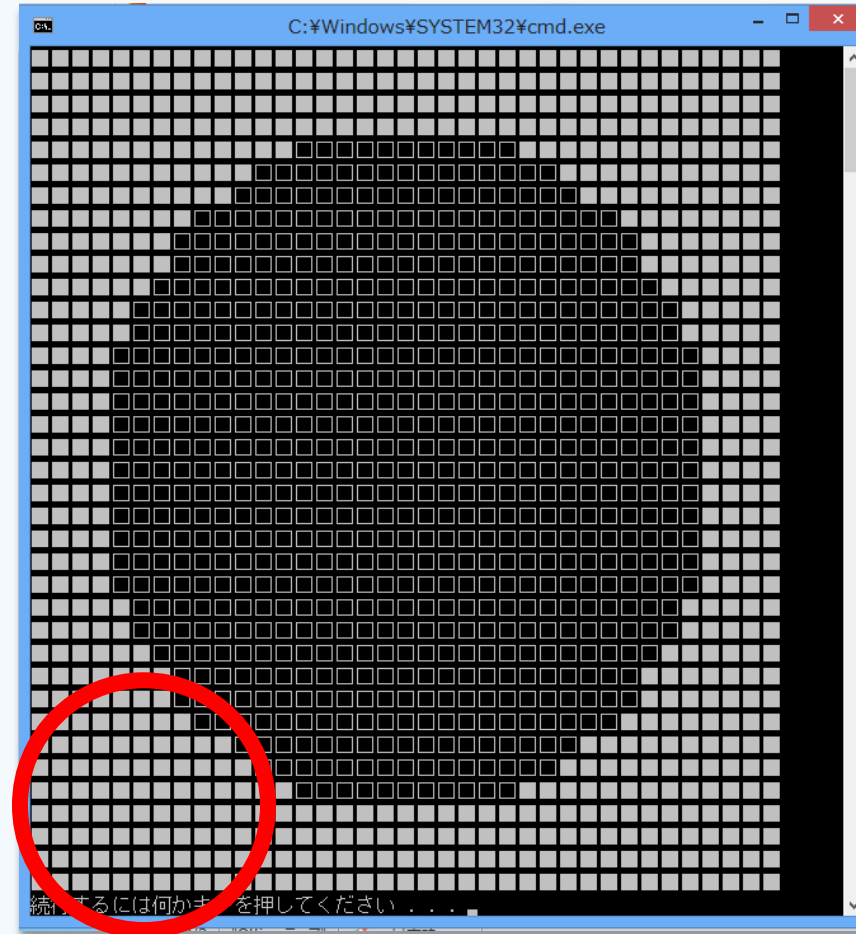
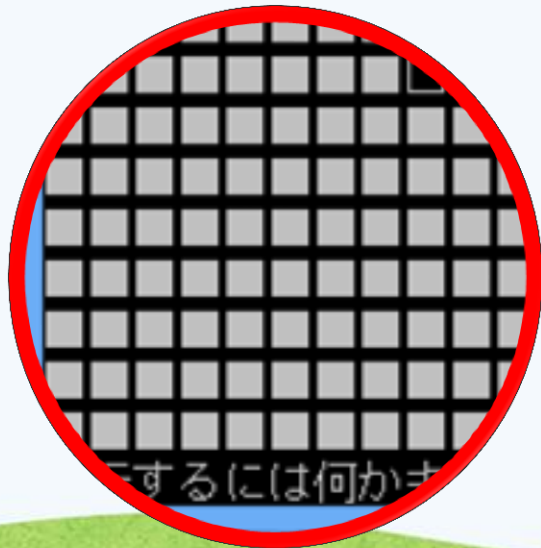




# [問題] 円を描いてみよう

■と□で円を描いてみよう

```
printf("□");  
printf("■");
```





# 考え方のヒント

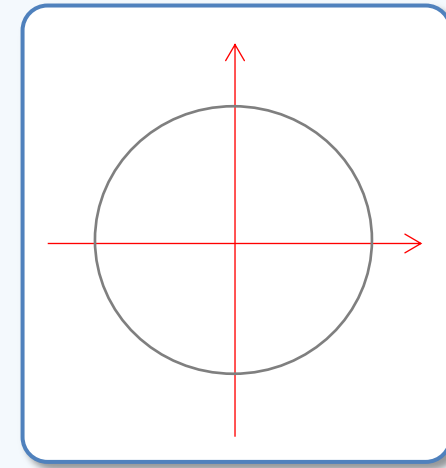
①円を横一列に10個描く

```
for(i = 1; i <= 38; i++)    printf("□");
```

②□と■の条件わけ

```
if(半径がrより小さい)    printf("□");  
else                        printf("■");
```

③x, yの座標のように考えてみよう



半径は $\sqrt{x^2 + y^2}$

これがrより小さいのだから...

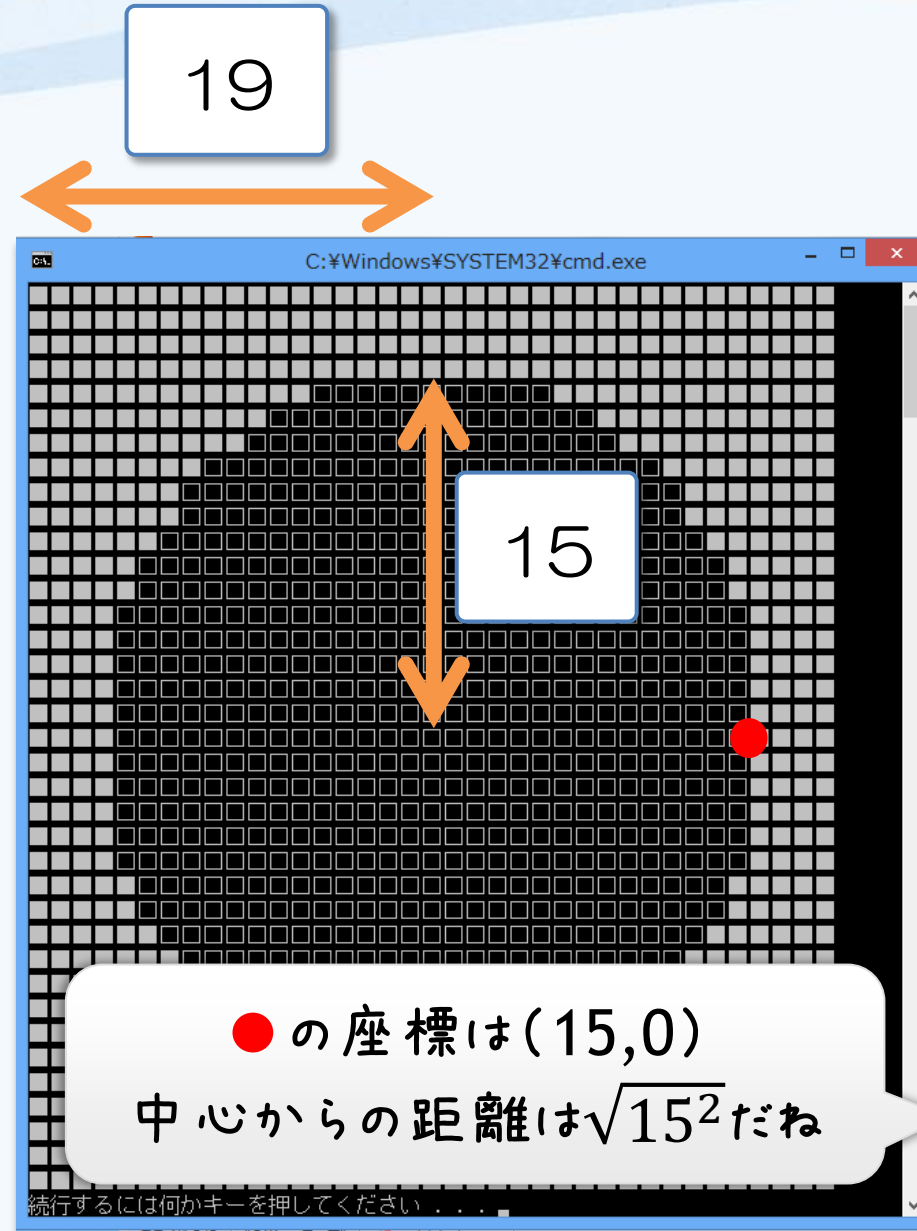


初期値

```
int r = 15; //半径  
int w = 18; //幅
```

```
int x; //x座標  
int y; //y座標
```

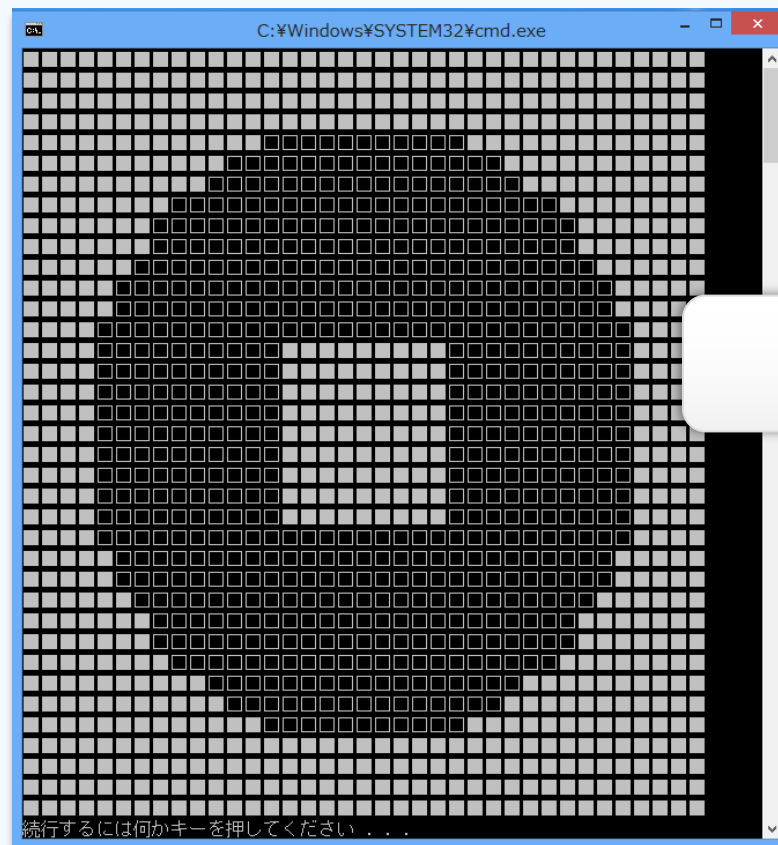
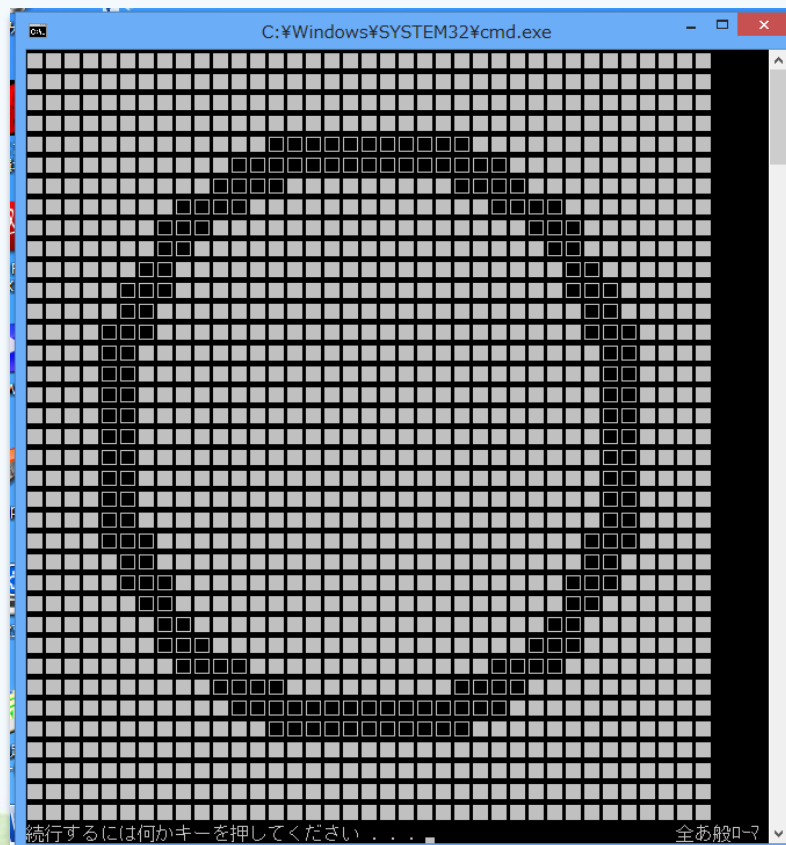
for文とif文を使えば書けるかな？



# 終わった人は・・・

ドーナツ型

お金？



わーい!!!



# こたえ

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

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

```
    int w = 18; //幅
```

```
    int r = 15; //半径
```

```
    int x, y;    //座標
```

```
    for(y = w; y >= -w; y--){
```

```
        for(x = -w; x <= w; x++){
```

```
            if(x*x + y*y < r*r){  
                printf("□");
```

```
            }
```

```
            else{  
                printf("■");
```

```
            }
```

```
        }  
        printf("¥n");
```

```
    }
```

```
    return 0;
```

```
}
```



```
#include <math.h>
```

```
if(sqrt(x*x + y*y) < r)
```

■ ドーナツ型 ■

```
if((x*x + y*y) < r*r){
```

```
    if(x*x + y*y < r2*r2){  
        printf("■");
```

```
    }
```

```
    else{  
        printf("□");
```

```
    }
```

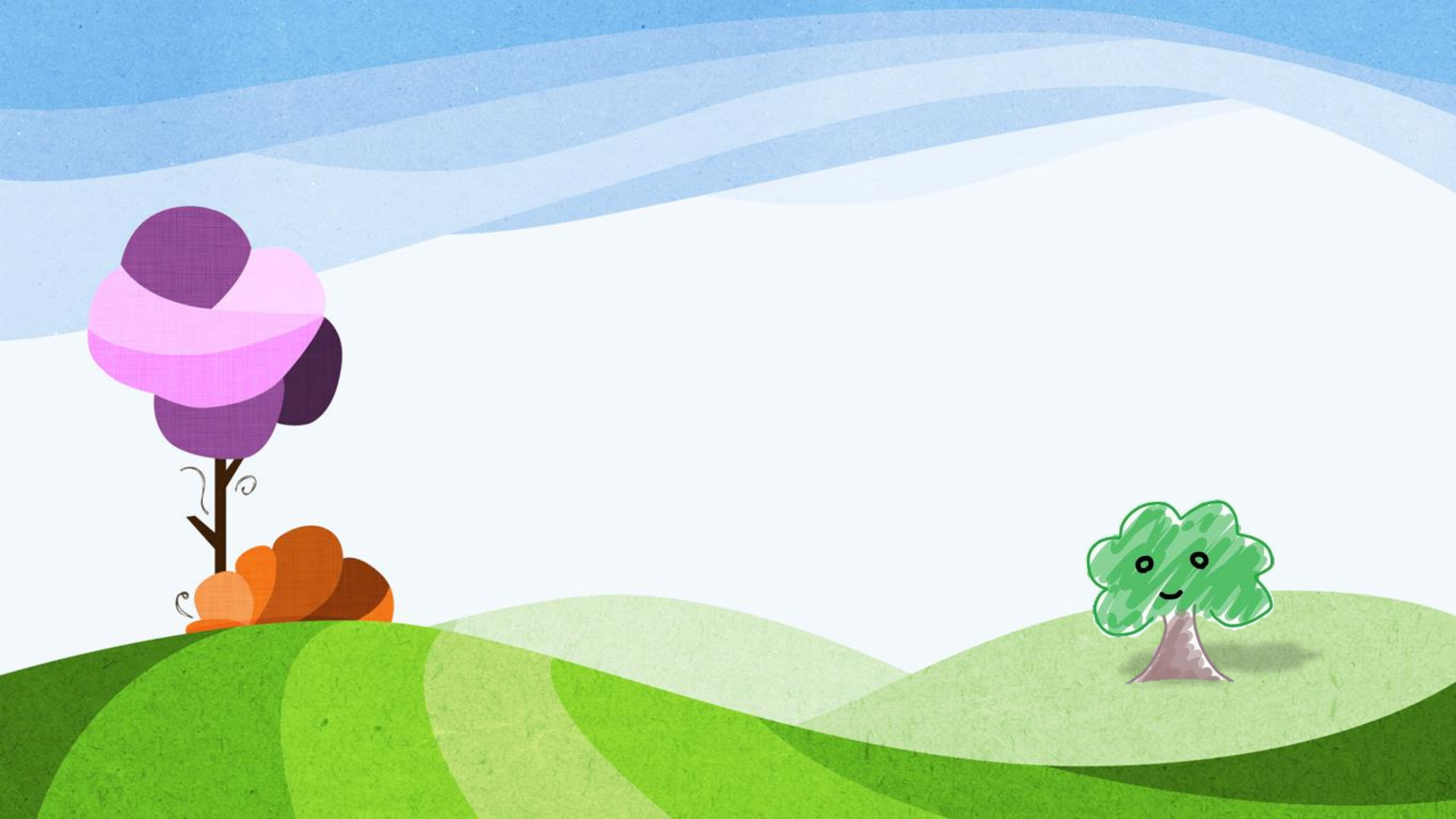
```
}
```

```
else{  
    printf("■");
```

```
}
```







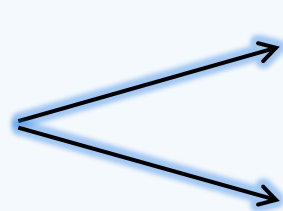
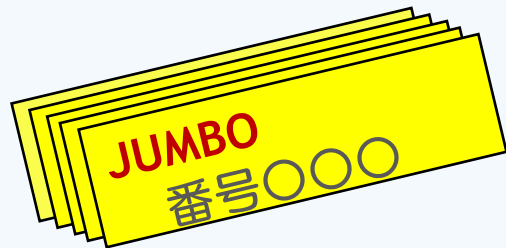
## 【問題】宝くじの当選金額

おーく君が5枚の宝くじを買いました。くじの番号はそれぞれ  
**{101,305,222,571,496}**でした。番号が偶数の時100円、素数の時1000円もら  
えたとしたら、おーく君の当選金額は合計いくらになるでしょう？

①偶数、素数を判定する関数をそれぞれ作ろう！

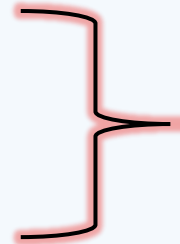
※引数を与えた時判定が真なら1、偽なら0を戻り値として返す。

②配列に番号を格納して、順番に当選金額を調べてみよう！



偶数？

素数？



いくら  
もらえるのかな？



# 【ヒント】宝くじの当選金額 ～①関数を作ろう！～

```
/*偶数かどうか判定する関数*/  
int even(int a){  
    int flag; //結果の値  
    if( )  
        flag = 1; //偶数の場合  
    }else {  
        flag = 0; //偶数でない場合  
    }  
    return flag; //計算結果を返す  
}
```

```
/*素数かどうか判定する関数①*/  
int prime(int a){  
    int i; //カウント変数  
  
    for(i= i< i++){  
        if( )  
            return 0; //素数でない場合  
        }  
    }  
    return 1; //素数の場合  
}
```

※偶数→2で割り切れる整数

※素数→1 と自分自身以外に正の約数を持たない数

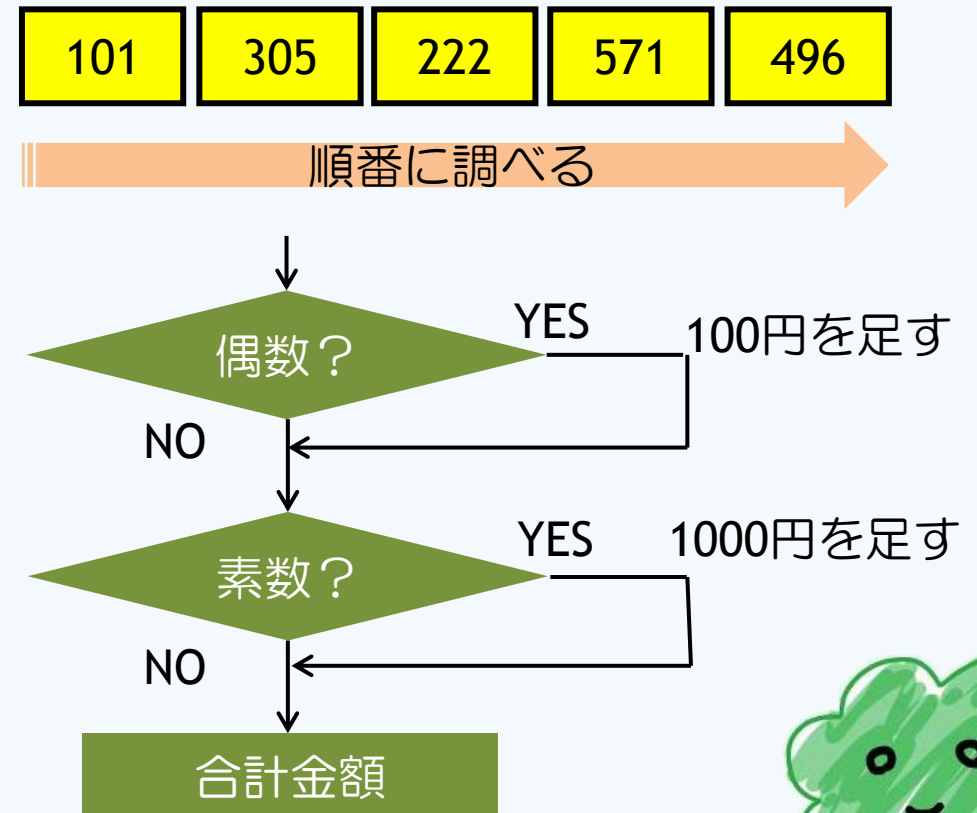
長いプログラムを書くときはこまめにデバッグしようね





# 【ヒント】宝くじの当選金額 ～②main()関数を作ろう！～

```
int main(void){  
    /*変数・配列の定義, 初期化など*/  
    int ans = 0, i;  
    int number[] = {101, 305, 222, 571, 496};  
    /*for文を使って配列の中身を順番に調査*/  
    for (i = 0; i < 5; i++)  
    {  
        if (number[i] % 2 == 0) //偶数かどうか?  
        {  
            ans += 100; //当選金額の計算  
        }  
  
        if (is_prime(number[i])) //素数かどうか?  
        {  
            ans += 1000; //当選金額の計算  
        }  
    }  
  
    /*合計金額を出力*/  
    printf("合計金額は%d円です。", ans);  
  
    return 0;  
}
```



## 【答え】宝くじの当選金額

```

// 関数定義
int main(void){
    // 変数宣言
    int a = 0, b;
    int number[] = {100, 200, 300, 400, 500};
    // 変数初期化
    // 配列の要素を順番に初期化
    for(i = 0; i < 5; i++){
        if (even(number[i]) == 1){ // 偶数の場合
            a += 100; // 100を足す
        }
        if (odd(number[i]) == 1){ // 奇数の場合
            b += 1000; // 1000を足す
        }
    }
    // 結果を出力
    printf("偶数の和は%d, 奇数の和は%d\n", a, b);

    return 0;
}

```

```

int main()
{
    int a = 10;
    int b = 20;
    int c = 30;
    int d = 40;
    int e = 50;
    int f = 60;
    int g = 70;
    int h = 80;
    int i = 90;
    int j = 100;
    return 0;
}

```

```

// Returns the sum of all elements in the array
int getSum(int arr[])
{
    int sum = 0;

    for(int i = 0; i < arr.length; i++)
    {
        sum += arr[i];
    }

    return sum;
}

```



合計金額は2200円です





# 【おまけ】 計算量について

```
for(i=2; i<=a-1; i++){  
    if(a%i == 0){  
        return 0; //素数でない場合  
    }  
}  
return 1; //素数の場合
```

除算を行った回数**671**回

```
if((even(a))==1&&a!=2) return 0; //2以外の偶数の時  
for(i=3; i<=a-1; i+=2){  
    if(a%i == 0){  
        return 0; //素数でない場合  
    }  
}  
return 1; //素数の場合
```

除算を行った回数**339**回

```
for(i=2; i*i<=a-1; i++){  
    if(a%i == 0){  
        return 0; //素数でない場合  
    }  
}  
return 1; //素数の場合
```

除算を行った回数**37**回

もっと計算量を  
減らせるかな?





おしまい