

第8回演習の確認

- ・プログラムの基本構造を理解した。
- ・宣言、変数、算術式、for ループ、書式付き出力(sprintf)を学習した。(華氏摂氏温度変換 Prg)

第9回演習の目標

- ・実数計算、記号定数について学習する。
- ・繰り返し計算によって、計算速度を計測しよう (time コマンドを使用)。

1・温度変換を実数で計算する。

```
main()
{
    int    fahr;
    printf(" 華氏    摂氏  %n");
    for(fahr=0;fahr <= 300;fahr=fahr + 20)
        printf("%3d%6.1f\n", fahr, 5.0/9.0*(fahr-32));
}
```

↓
celsius の代わりに数式を書いている。
全体6桁、小数点以下1桁の小数点表示

2・記号定数

上記のプログラムには、0、300や20など意味のはっきり分からない数値(マジック・ナンバー)が存在する。これらの数値の意味が分かるように、記号名で定義(記号定数)しておくと以下のメリットがある。記号定数は大文字で書くのが一般的である。

- メリット
- ・プログラムが読みやすくなる
 - ・ “ ” のメンテナンスが容易になる

```
#define LOWER    0
#define UPPER    300
#define STEP      20
```

/* ← LOWER に 0 を割り当てる */

記号定数の定義
= ; がないので注意

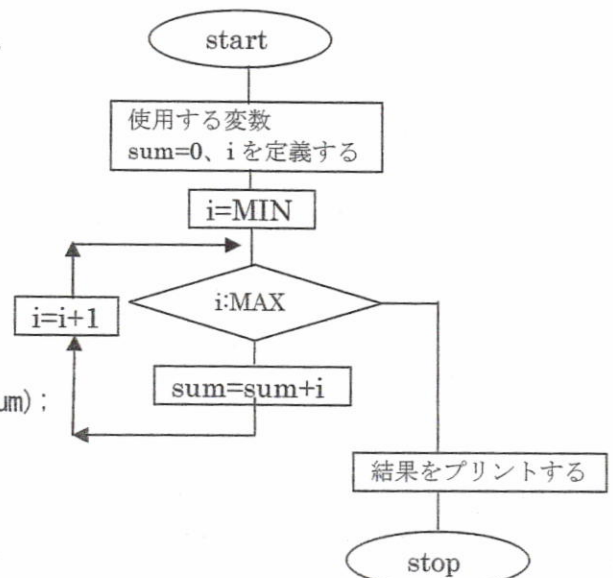
コメント：プログラムが読みやすくなるので積極的に使うこと

```
main()
{
    int    fahr;
    printf(" 華氏    摂氏  %n");
    for(fahr=LOWER;fahr <= UPPER;fahr=fahr + STEP)
        printf("%3d%6.1f\n", fahr, 5.0/9.0*(fahr-32));
}
```

3・記号定数を使い、大量の繰り返し計算をしてみよう。

```
#define MIN 1
#define MAX 100000
main()
{
    float sum=0.0;
    int i;
    for(i=MIN;i <= MAX;i=i + 1)
        sum = sum + i;
    printf("%d から%d までの合計は%f です。 %n", MIN, MAX, sum);
}
```

↑
浮動小数点数



【課題】 1億までの和を求めて処理時間を調べてみよう。

【演習】

- (1) 1 から 1 億までの和を求める。(sumloku.c)
 (2) プログラムが完成したら、処理時間を求める。(time ./a.out)

4・素数とは1と自分自身以外には約数を持たない自然数である。素数を表示してみよう。

```
#define MIN 2
#define MAX 1000
main()
{
    int i, j;

    for(i=MIN; i <= MAX; i++) {
        for(j=2; j < i; j++)
            if(i%j == 0) break;
        if(i - j == 0)
            printf("%d\n", i);
    }
}
```

上のプログラムを作り、実行してからスライドのフローチャートで説明する。

- 【課題】 (1) 10000000までの、素数をプリントする。
 (2) 上記素数を、1000個ごとにプリントする。

ヒント

- ① 素数の数を数えるために、新しいカウンター (k) を定義する。
 int k=0;
 ② 素数が見つかったら、kに1たす。
 k=k+1;
 ③ プリントする前に k をチェックする。
 k を1000で割ってみて、割り切れたらプリントする。
 割り切れなかったらプリントしない。

