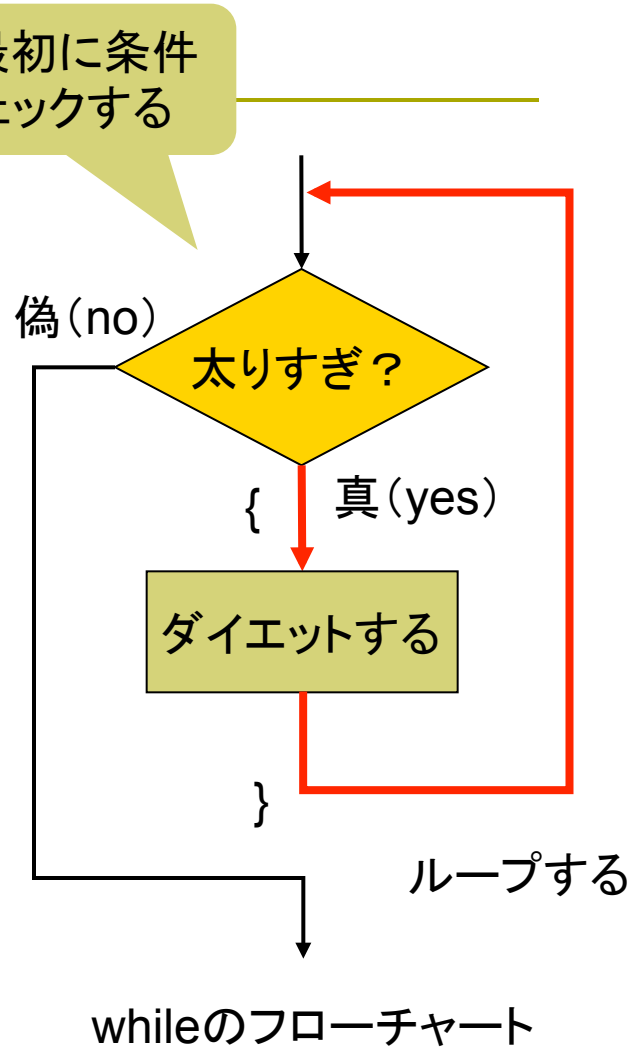
# Programming II 0x03

配列（2011.04.28）

塩澤秀和 http://vilab.org

# while文（復習）

- 繰り返し（ループ、反復）
  - 継続する条件が成り立つ間、決められた処理を何回も繰り返す
  - “while”＝「～である間」
  - 条件が不成立なら、処理を飛ばす



whileのフローチャート

- while文の構文

```
while (条件式) {
   条件が“真”の間、繰り返す処理
   …
}
```

- **注意：** 条件式は、終わる条件でなく、**“続ける条件”**を書く

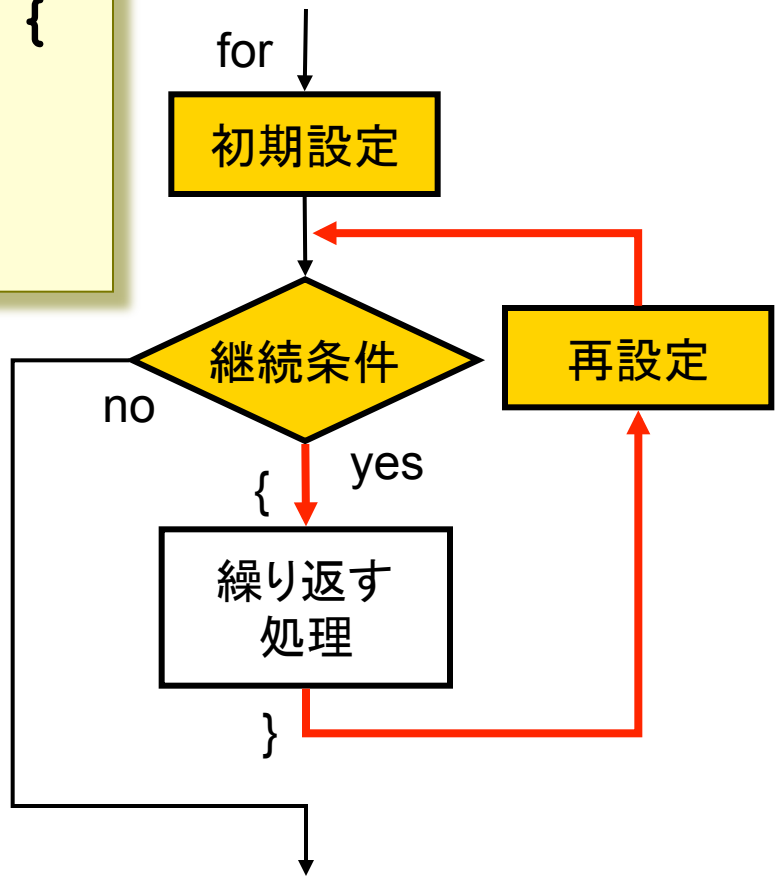
# for文（復習）

よくあるループのパターンを
簡潔に書くための構文

- for文の構文

```
for （初期設定； 継続条件； 再設定） {
        条件が“真”の間、繰り返す処理
        …
}
```

- while文で書き換えてみよう




forのフローチャート

フローチャートを
書いてみよう

# for文の例（復習）

```
#include <stdio.h>

int main(void)
{
  int i;
  int n, sum = 0;

  printf("整数10個？\n");

  for (i = 0; i < 10; i++) {
    scanf("%d", &n);
    sum += n;
  }

  printf("合計 %d\n", sum);
  return 0;
}
```

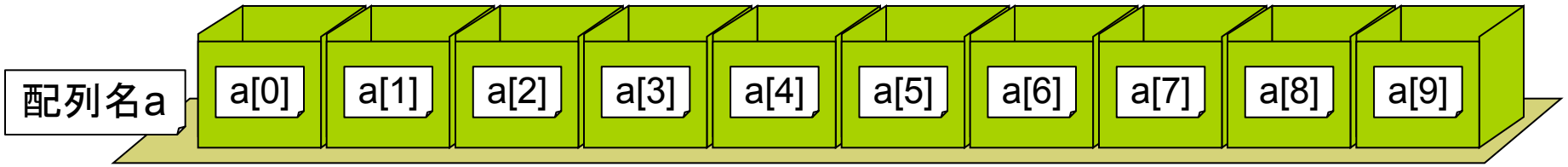
総和（合計）は
最初０に設定

総和にどんどん
数を足していく

# 配列変数

- 配列（array）とは
  - 同じデータ型の変数をまとめて作る
  - 各変数（**要素**）には、配列名と番号（**添字**）でアクセスする

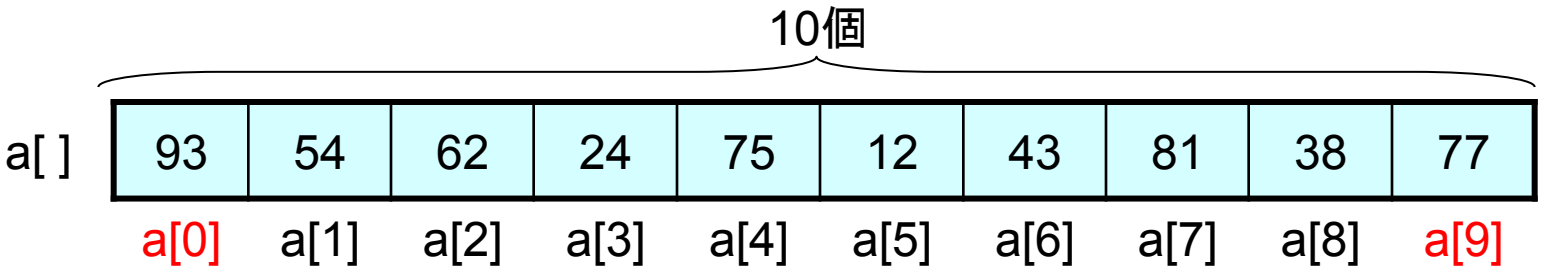
配列のイメージ（要素数10個）




- どんなときに使うか

# 配列の作りかた

- 配列の定義
  - 定義(宣言): 要素の型 配列名 [要素数]; (p.101)
    int a[10]; ← a[0] ～ a[9] の10個の変数をまとめて作る
  - 初期化(p.104)
    int a[10] = { 93, 54, 62, 24, 75, 12, 43, 81, 38, 77 };




- 配列の要素
  - 配列の要素(a[0]など)は、単独の変数と同じように使える
  - [ ] 内の添字(そえじ index)は、**0から始まる番号**(整数)
  - **【注意】** int a[10] と定義した配列には a[10] は**ない!!**

# 配列要素の参照

```
#include <stdio.h>

int main(void)
{
  int index;

  double a[3];

  a[0] = 0.1;
  a[1] = 0.2;
  a[2] = 0.3;

  printf("番号(0～2)？ ");
  scanf("%d", &index);

  printf("a[%d] = ", index);
  printf("%f\n", a[index]);
  return 0;
}
```

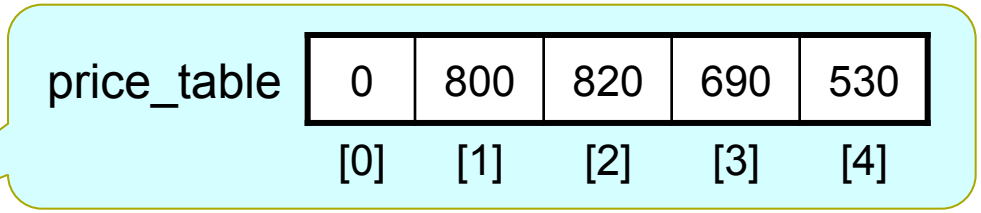
この部分を…

配列要素の初期化で書き換えるとどうなるか？

scanfで読み込むように書き換えるとどうなるか？

【ヒント】a[0]やa[1]はdouble型の変数と同じ

# 「番号→データ」の表




```
#include <stdio.h>


// 価格表(グローバル変数)
int price_table[] = {
  0, 800, 820, 690, 530 };

int main(void)
{
  int order, price;

  printf("定食メニュー\n");
  printf("1. 生姜焼き\n");
  printf("2. とんかつ\n");
  printf("3. さんま\n");
  printf("4. お子様ランチ\n");
  printf("定食の番号？ ");
  scanf("%d", &order);
  if (order < 1 || 4 < order) {
    printf("入力エラーです\n");
    return 1;  // プログラム終了
  }

  // 価格表で「番号→価格」の変換
  price = price_table[order];

  printf("%d円です\n", price);
  return 0;
}
```

# 配列とループ

フローチャートを書いてみよう

```
#include <stdio.h>

int main(void)
{
  int data[10];
  int i;

  printf("10個の整数を入力\n");
  for (i = 0; i <= 9; i++) {
    scanf("%d", &data[i]);
  }

  printf("偶数だけを再表示\n");
  for (i = 0; i <= 9; i++){
    if (data[i] % 2 == 0) {
      printf("%d\n", data[i]);
    }
  }
  return 0;
}
```

条件があてはまる要素だけをif文でピックアップする

# ループと関数

```
// 整数のデータを5つ読み込んで
// 横型の棒グラフにして表示する
#include <stdio.h>

// n個の■を表示する関数
void printbar(int n)
{
  printf("%2d|", n);
  while (n > 0) {
    printf("■");
    n--;
  }

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

ループの動作を理解せよ

```
int main(void)
{
  int data[ 5 ];
  int i;

  printf("データ入力\n");
  for (i = 0; i < 5 ; i++) {
    printf("data[%d] = ", i);
    scanf("%d", &data[i]);
  }

  printf("\n");
  for (i = 0; i < 5 ; i++) {
    printbar(data[i]);
  }
  return 0;
}
```

データの個数を変えたいとき、書き換える場所は…

i<=4 より i<5 のほうが一般的

# マクロ定数（復習）

- プリプロセッサとは（p.93）
  - 前処理（コンパイルする前の処理）をするソフトウェア
  - 「#」で始まるプリプロセッサ指令に従って、ソースコードを“加工”する
- #define 指令（p.95）
  - 書式：「#define　マクロ名　文字列」
  - ソースコード内のマクロ名を、すべて文字列で置き換える
    例：#define PI 3.14　← 以後、「PI」と書くと「3.14」と同じ意味になる
  - さらに高度な #define ⇒ 引数付きマクロ（p.97）
- マクロの利点
  - 定数に名前をつけると、プログラムの意味が分かりやすくなる
  - プログラムを修正するとき、1ヶ所の定義を変更すればよい
  - 単なる文字列の置き換えなので、変数のようにメモリを使わない

# 2つの配列の和

```
// 2つのN次元ベクトルの和
#include <stdio.h>
// マクロ定数 N = 3 の定義
#define N 3

int main(void)
{
  // 同じ大きさの配列を3つ作る
  double v1[N], v2[N], v3[N];
  int i;

  // v1, v2をキーボードから読み込む
  for (i = 0; i < N; i++){
    printf("v1[%d] = ", i);
    scanf("%lf", &v1[i]);
  }
  for (i = 0; i < N; i++){
    printf("v2[%d] = ", i);
    scanf("%lf", &v2[i]);
  }

  // 要素ごとに和を計算する
  for (i = 0; i < N; i++){
    v3[i] = v1[i] + v2[i];
  }

  // 結果v3を表示する
  for (i = 0; i < N; i++){
    printf("v3[%d] = %f\n",
           i, v3[i]);
  }
  return 0;
}
```

配列のサイズを変えたいときにはここだけ変えればいい

# 演習問題

3a.「配列とループ」のプログラムのフローチャートを書きなさい。

- この問題の解答はアップロードしなくてよい。

3b. キーボードから10個の整数を読み込んだ後、入力とは逆の順番で表示するプログラムを作成しなさい。

3c. int型の配列を { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 } で初期化し、全要素の合計と平均を表示するプログラムを作成しなさい。

3d. AさんとBさんの今後1週間の予定を、それぞれ int a[7] と int b[7] に読み込み、2人とも予定がない日をすべて表示するプログラムを作成しなさい（詳細は各自で設計すること）。

- 次回までの課題：教科書p.105～120 を予習（入力＆実行）