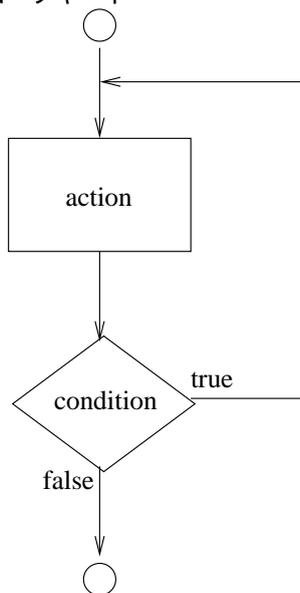


4.6 do/while 反復構造

- do/while 反復構造
 - while 構造に類似
 - ループの本体を実行した後で継続条件をテストする
 - * ループ本体は少なくとも 1 回は実行される
 - フローチャート



- 書式 (この書き方は通常使わない)

```
do
    文
while (条件式);
```
- 一般的な書き方: プログラムを見易くするために次のように書く

```
do {
    [複数の] 文
} while (条件式);
```
- 例: 1 から 10 の整数を表示

```
int counter = 1;
do {
    printf("%d ", counter);
} while (++counter <= 10);
```

4.7 break 文と continue 文

4.7.1 break

- while, for, do/while, switch 構造からただちに抜け出す
 - break 文の一般的利用法
 - ループを途中で抜け出す
 - switch 構造の残りの部分をスキップする
- 注意) 入れ子になった反復構造から一気に抜け出すことはできない

リスト 4.6 for 構造における break 文の使用例

```
/* for 構造における break 文の使用例 */
#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 10; i++) {

        if (i == 5)
            break; /* i が 5 のときだけ、ループから抜け出す */

        printf("%d ", i);
    }

    printf("\nループを抜け出したときの i は %d\n", i);

    return 0;
}
```

実行結果

```
1 2 3 4
ループを抜け出したときの i は 5
```

4.7.2 continue

- while, for, do/while 構造の本体の残りをスキップし、次の反復に移る
- while と do/while 構造では
 - continue を実行すると、ただちにループ継続条件がテストされる

- for 構造では
 - continue を実行すると、インクリメント式 (for の式 3) が実行され、ループ継続条件 (for の式 2) がテストされる

リスト 4.7 for 構造における continue 文の使用例

```

/* for 構造における continue 文の使用例 */
#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 10; i++) {

        if (i == 5)
            continue; /* i が 5 のときだけ、ループ内の残りの文をスキップする */

        printf("%d ", i);
    }

    return 0;
}

```

実行結果

```
1 2 3 4 6 7 8 9 10
```

4.7.3 浮動小数点数と break の利用例

リスト 4.8 浮動小数点数と break を使った累乗プログラム

```

/* 実数 x の n 乗を求める入力エラー処理付きプログラム */
#include <stdio.h>

int main()
{
    /* 浮動小数点数 (実数) を格納できる変数 x と power の宣言 (float 型) */
    float x, power;
    int i, n;

    printf("Enter real number: ");
}

```

```

scanf("%f", &x); /* float 型の値を scanf で入力するには %f を使う */

while (1) {
    printf("Enter non-negative integer: ");
    scanf("%d", &n);
    if (n >= 0)
        break;
    else
        printf("Input Error\n");
}

power = 1;
for (i = 0; i < n; i++) {
    power *= x;
}

/* float 型の値を printf で出力するには %f を使う */
printf("%f^%d = %f\n", x, n, power);

return 0;
}

```

実行結果例

```

Enter real number: 0.3
Enter non-negative number: -1
Input Error
Enter non-negative number: 5
0.300000^5 = 0.002430

```