第三回 ループ文の例

1. 初めに

あまりに例の量が多すぎるので,解説は少なめである.ソースコードと実行結果を照らし合わせて,プログラムの流れを理解してほしい.

1. 例題

[幅w,高さhの四角形を表示する]

|  |
| --- |
| #include <stdio.h>  int main(void) {      int w, h;      int i, j;      scanf("%d %d", &w, &h);      for(i = 0; i < h; i++) {          for(j = 0; j < w; j++) {              printf("\*");          }          printf("\n");      }      return 0;  } |

実行結果(16, 5が入力例)

|  |
| --- |
| 16 5  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

[幅w,高さhの四角形を表示する2]

|  |
| --- |
| #include <stdio.h>  int main(void) {      int w, h;      int i, j;      scanf("%d %d", &w, &h);      for(i = 0; i < h; i++) {          for(j = 0; j < w; j++) {              if((i + j) % 2 == 1) printf("+");              else printf("\*");          }          printf("\n");      }      return 0;  } |

実行結果(16, 5が入力例)

|  |
| --- |
| 16 5  \*+\*+\*+\*+\*+\*+\*+\*+  +\*+\*+\*+\*+\*+\*+\*+\*  \*+\*+\*+\*+\*+\*+\*+\*+  +\*+\*+\*+\*+\*+\*+\*+\*  \*+\*+\*+\*+\*+\*+\*+\*+ |

if文で\*を表示するか,+を表示するかを分岐している.

[高さhの直角二等辺三角形を出力する]

|  |
| --- |
| #include <stdio.h>  int main(void) {      int h;      int i, j;      scanf("%d", &h);      for(i = 0; i < h; i++) {          for(j = 0; j < i + 1; j++) {              printf("\*");          }          printf("\n");      }      return 0;  } |

実行結果(8が入力例)

|  |
| --- |
| 8  \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\* |

[高さhの四角形を表示する2]

|  |
| --- |
| #include <stdio.h>  int main(void) {      int h;      int i, j;      scanf("%d", &h);      for(i = 0; i < h; i++) {          for(j = 0; j < h - i - 1; j++)              printf(" ");          for(j = 0; j < i + 1; j++)              printf("\*");          printf("\n");      }      return 0;  } |

実行結果(8が入力例)

|  |
| --- |
| 8  \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\*  \*\*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*\* |

1. 問題を解くときによく使うと思われる例

何かの問題を解くときに使う定型文のようなものを紹介する.

もちろん下の例を丸ごと使うわけでなく,一部を改変したりしてうまく問題に合わせて利用する.ソースコードの内容を理解して,自分で書けるようになろう.

[入力された10個の整数から最大値,最小値を見つける]

入力された整数値から最大値,最小値を見つける.今回は0以上10000000以下の値に限定しておく.

以下に示す方法は,挑戦者が王者を討つ戦いだと思えばよい.初めに,王者をとても弱く設定しておく.そのあと,挑戦者とたたかわせて,挑戦者が勝てば次の王者となる.

最大値を求める場合は,まず非常に小さい値をmaxに設定しておき,順に対象となる値と比較していく.値がmaxより大きいなら,それを新たなmaxに設定する.

最小値の場合も同様.初めに大きい値をminに設定しておき,順に対象となる値と比較していく.値がminより小さいなら,それを新たなminに設定する.

|  |
| --- |
| #include <stdio.h>  int main(void) {  int a[10];  int i;  int max = -1, min = 10000000;  printf("input: ");  for (i = 0; i < 10; i++) {  scanf("%d", &a[i]);  }  for (i = 0; i < 10; i++) {  if (max < a[i]) {  max = a[i];  }  if(min > a[i]) {  min = a[i];  }  }  printf("max: %d, min: %d\n", max, min);    return 0;  } |

[入力されたN個の整数の総和を求める]

総和を格納しておく変数sumに,整数の値を足していけばよい.sumを0で初期化しておくことを忘れないように.

|  |
| --- |
| #include <stdio.h>  int main(void) {  int a[1000];  int N, i, sum = 0;  scanf("%d", &N);  for (i = 0; i < N; i++) {  scanf("%d", &a[i]);  }  for (i = 0; i < N; i++) {  sum += a[i];  }  printf("%d\n", sum);  return 0;  } |

[文字列走査の基本]

継続条件に「今見ている文字はヌル文字でない」という条件を書くことで,文字列の先頭から末尾までを走査することができる.以下は,入力した文字列の文字を先頭から順に改行付きで出力し,最後にその文字数を表示するプログラムである.

|  |
| --- |
| #include <stdio.h>  int main(void) {      char str[100];      int N;      scanf("%s", str);      for (N = 0; str[N] != '\0'; N++) {          printf("%c\n", str[N]);      }      printf("%d\n", N);      return 0;  } |

さて,文字列の走査を例で示したが,文字数を数えるだけのためにわざわざfor文を書くことはない.Cでは,文字数を数える関数strlenが用意されているからである.

|  |
| --- |
| strlen(文字列) |

これを使うためには,ソースコード上方に以下の記述をする必要がある.

|  |
| --- |
| #include <string.h> |

先に挙げた例は次のように書き換えられる.

|  |
| --- |
| #include <stdio.h>  #include <string.h>  int main(void) {  char str[100];  int N;  scanf("%s", str);  N = strlen(str);  printf("%d\n", N);  return 0;  } |

[各桁の値の和を求める]

|  |
| --- |
| #include <stdio.h>  int main(void) {      int N;      int sum = 0;      scanf("%d", &N);      while(N) {          sum += N % 10;          N /= 10;      }      printf("%d\n", sum);      return 0;  } |

N % 10でNの一桁目を取り出し,Nを10で割ることで桁を右にずらす.

Nが0になれば,whileを抜ける.

例えばN = 123のとき,次のように動く.

N % 123 → 3をsumに加え,Nを10で割ってN = 12とする.

N % 12 → 2 をsumに加え,Nを10で割ってN = 1とする.

N % 1 → 1をsumに加え,Nを10で割って N = 0とする.

N が0なのでループを抜ける.

[多重ループを抜ける]

以下は,掛け算九九表を出力するプログラムだが,70以上の値が出てきた時点で表示を切りやめる.

|  |
| --- |
| #include <stdio.h>  int main(void) {      int i, j;      int flag = 0;      for (i = 1; i <= 9; i++) {          for (j = 1; j <= 9; j++) {              int p = i \* j;              if (p >= 70) {                  flag = 1;                  break;              } else {                  printf("%3d", i \* j);              }          }          printf("\n");          if(flag) break;      }        return 0;  } |

iとjの積を変数pで管理して,pが70以上のとき,flagの値を1にして一つのループから抜ける.その外側には「flagが1かどうか」を問うif文がある.もしそれが真ならbreakでさらにもう一つのループから抜けるようになっている.「積の値が70以上になった」ことを一つ外側のループに伝えるために,flagという変数が使われている.

このように,何かの出来事が「起こらなかった」「起こった」を0と1で管理する変数を「フラグ」という.

[無限ループとその終了方法]

以下は無限ループをするプログラムである.

|  |
| --- |
| #include <stdio.h>  int main(void) {        while(1) {          printf("HAHAHA!!!\n");      }      return 0;  } |

whileの継続条件が常に1のため,HAHAHA!!!という文字列がひたすら出力されて,プログラムが自動で終了することはない.

何かのプログラムを書いていた時,継続条件や計算の間違いによって,無限ループのバグに陥ってしまうことはよくある.

**[Ctrlキー] + [cキー]でプログラムを強制終了できる**ので,もしものときのために覚えておいてほしい.

[while(1)をあえて指定して,if文でループを抜ける]

whileは継続条件を見る位置がループの先頭のみである.ループの途中で抜ける条件を作りたいときは,if文とbreak文をうまく利用する.

以下は,二つの正整数を入力して,その和を表示するプログラムである.ただしどちらか一方に負の数を入力するとプログラムが終了する.

|  |
| --- |
| #include <stdio.h>  int main(void) {      int a, b;      while (1) {          scanf("%d %d", &a, &b);          if(a < 0 || b < 0) break;          printf("%d\n", a + b);      }      return 0;  } |

このように繰り返し入出力をする形式の問題はAOJで出てくるので覚えておこう.