

1 演習課題 1(デバッガ機能を使ってみる)

まず、Code::Blocks で新規プロジェクトを作成し、main.c をリスト 1 のように変更しなさい。つぎに、デバッグ (D) デバッグウィンドウ 監視 (図 1 の左) をクリックしなさい。そして、以下をしなさい。

1. カーソルを 1 つ目の printf() の行 (8 行目) にしてから、「カーソル位置まで実行 (Run to cursor)」のボタン (図 2) を押す
2. Watches ウィンドウ (図 1 の右) の Local variables を開く
3. 「次の行 (Next line)」ボタンを押して 1 行ずつ順に実行されていく様子を見る

上記以外に、デバッグボタンの動作をいろいろ試してみることに。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int a = 0, b;
7
8     printf("Hello world(1)!\n");
9     a + 1;
10    printf("Hello world(2)!\n");
11    b = a + 1;
12    a ++;
13    b --;
14    printf("Hello ");
15    printf("world(3)!\n");
16    a = 2;
17    printf("Hello world(%d)!\n", a+2);
18    printf("Hello world(5)!\n");
19    return 0;
20 }
```

リスト 1 (list-p1.c)

2 演習課題 2

- (1) リスト 1 を 8 行目から 1 行ずつ実行し、実行される行の番号、変数 a, b の値、表示される文字列を、プログラム終了まで記録しなさい。式と変数の値の変化、表示の関係について、よく観察すること。

そして、変数 a, b の初期値、9, 11, 12, 17 行目の実行に伴う変数の変化 (無変化) と表示について考察しなさい。

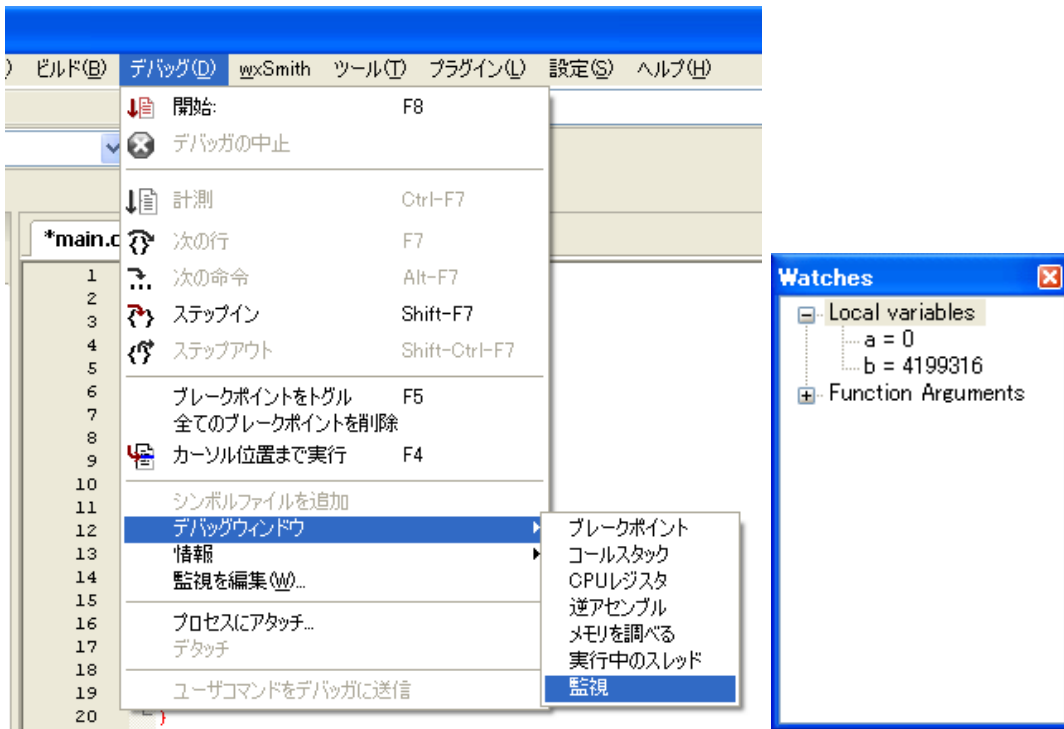


図 1: 左)Watches (監視) ウィンドウを開く, 右)Watches ウィンドウ

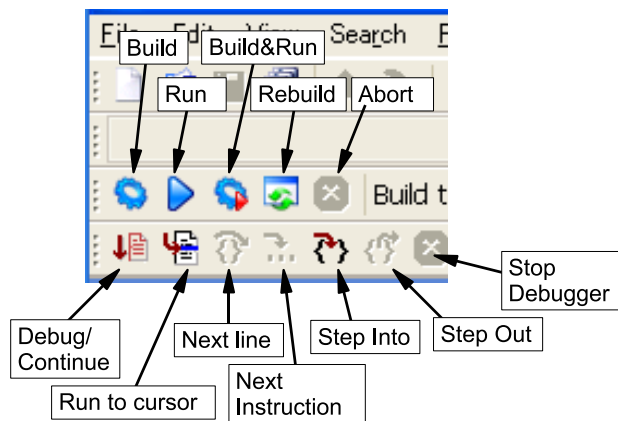


図 2: 実行とデバッグのボタン。

- (2) リスト 2 を読み、どのような表示が得られるか、まず予想しなさい。リスト 2 を 7 行目から 1 行ずつ実行し、実行される行の番号、変数 `sw` の値、表示される文字を、プログラム終了まで記録しなさい。そして、`case` 文と `break` 文の役割について考察しなさい。
- (3) リスト 3 を読み、どのような表示が得られるか、まず予想しなさい。リスト 3 を 28 行目から 1 行ずつ実行し、実行される行の番号、`main` 関数の `a` の値を、プログラム終了まで記録しなさい。「次の行 (Next line)」ボタンでなく「ステップイン (Step Into)」ボタンを使うこと。Watches ウィンドウの Function Arguments (関数の引数) も開き、観察すること。そして、特になぜ 10 行目が実行されないのか、28 行目と 31 行目の `test` 関数を呼んだ後の動作、`test3` 関数を呼んだ後の動作について考察しなさい。
- (4) リスト 4 を読み、どのような表示が得られるか、まず予想しなさい。つぎに 7 行目から 1 行ずつ実行し、実行される行の番号と表示される文字を、プログラム終了まで観察しなさい。16 から 25 行目と、42 から 49 行目を実行中については、1 行ずつ、行番号、変数 `i, j` の値と表示される文字を記録すること。そして、`for` ループと `break` 文、`continue` 文の役割について考察しなさい。

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int sw;
6
7     for (sw=0; sw<9; sw ++ ) {
8         switch (sw) {
9             case 0:      putchar('a'); break;
10            case 2:      putchar('b');
11            case 3:      putchar('c'); break;
12            case 4:
13            case 6:      putchar('d'); break;
14            case 1:      putchar('g'); break;
15            case 7:      break;
16            default:    putchar('X');
17                break;
18            }
19        }
20
21    return 0;
22 }

```

リスト 2 (list-p2.c)

```

1 #include <stdio.h>
2
3 int test(int b)
4 {

```

```

5  if (b > 0)
6      return b;
7
8  return 0;
9
10 b = b + 1;
11 return 1;
12 }
13
14 int test2(int b)
15 {
16     return b * 2;
17 }
18
19 int test3(int a, int b)
20 {
21     return test(a) + test(b);
22 }
23
24 int main()
25 {
26     int a;
27
28     a = test(10);
29     test(2);
30     a = a + test(10);
31     a = test(-5);
32     a = test2(a);
33     a = test3(10, 20);
34
35     return 0;
36 }

```

リスト 3 (list-p3.c)

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int i, j;
6
7     for (i=0; i<4; i++) {
8         for (j=0; j<4; j++) {
9             putchar('*');
10        }
11        printf(" \n");
12    }
13
14    printf("\n");
15

```

```

16  for (i=0; i<4; i++) {
17      for (j=0; j<4; j++) {
18          if (i == j) {
19              putchar(' ');
20              continue;
21          }
22          putchar('*');
23      }
24      printf(" \n");
25  }
26
27  printf("\n");
28
29  for (i=0; i<4; i++) {
30      for (j=0; j<4; j++) {
31          putchar('*');
32          if (i == j) {
33              putchar(' ');
34              continue;
35          }
36      }
37      printf(" \n");
38  }
39
40  printf("\n");
41
42  for (i=0; i<4; i++) {
43      for (j=0; j<4; j++) {
44          putchar('*');
45          if (i == j)
46              break;
47      }
48      printf(" \n");
49  }
50  return 0;
51 }

```

リスト 4 (list-p4.c)

3 付録

3.1 プロジェクトの新規作成

Code::Blocks でデバッグをするにはプロジェクトの作成が必要です。作成方法は以下のようにします。

1. Create a new project をクリックします (図 3)。
2. 出てきたウィンドウ (図 4) で、Console application を選び、Go をクリックします。

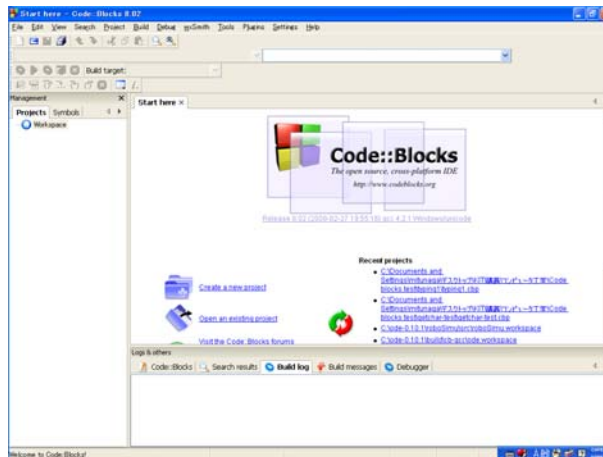


図 3: Code::Blocks の起動画面

3. 図 5 で Next をクリックします。
4. 使用する言語は「C」を選び、Next をクリックします (図 6)。
5. 図 7 で Project Title に「test」(何でも構いません) を入れ、Folder to create project in にソースファイルなどを入れるディレクトリを指定します。ここでは... をクリックして、c:\src\test を作りました。そして Next をクリックします。
6. 図 8 で Finish をクリックします。
7. プロジェクトが作成されるので、左の test 中の Sources から main.c をダブルクリックします (図 9)。リスト a が表示されます。デバッグボタン (図 2) を使って、実行してみましょう。

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     printf("Hello world!\n");
7     return 0;
8 }

```

リスト a (list-pa.c)

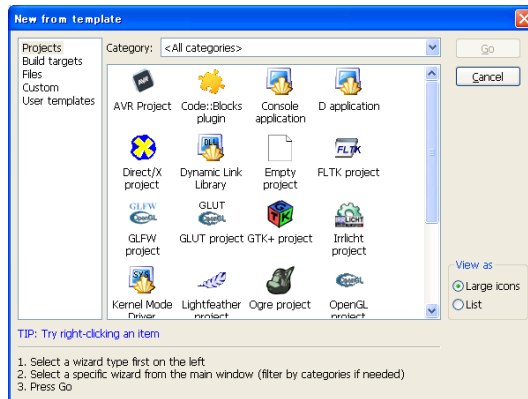


図 4: 作成するプロジェクトの種類を選ぶ

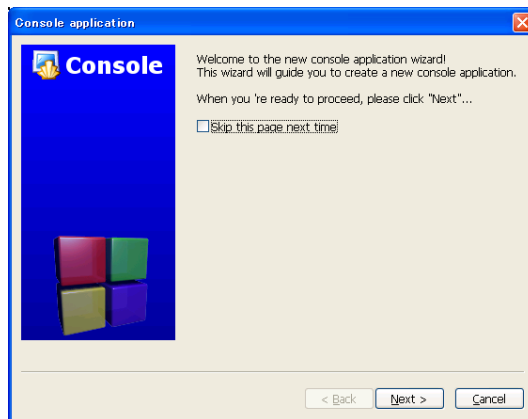


図 5: プロジェクト作成ウィザードの最初のウィンドウ

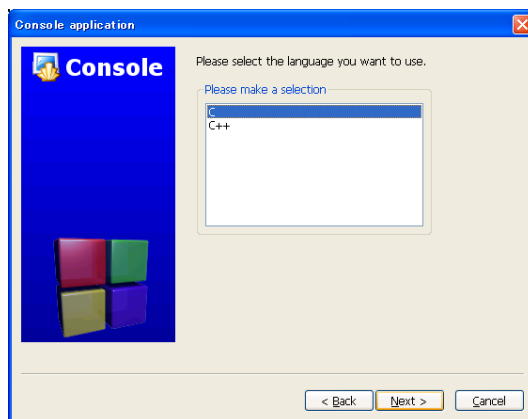


図 6: 使用する言語 (ここでは C または C++) を選ぶ

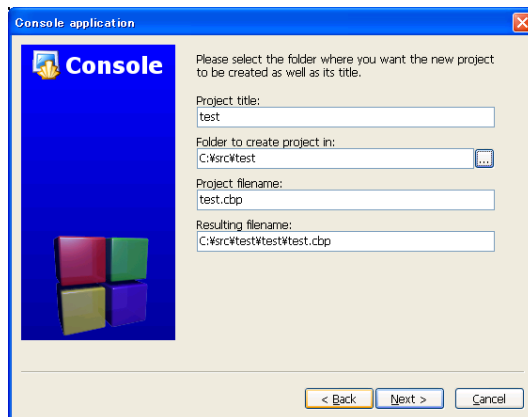


図 7: プロジェクトの名前とディレクトリを指定する

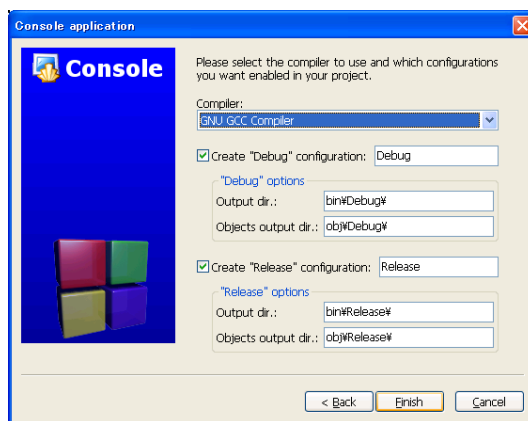


図 8: コンパイラ等を指定する。あらかじめ設定されている値でOK。

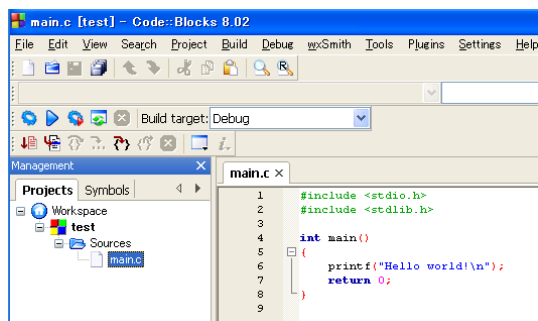


図 9: プロジェクトが作成されプログラムの雛形が用意されたところ。