

第 11 回課題 T11

(ポインタ I(アドレスとポインタ)：教科書第 12 章, 2008/6/26(木))

基本問題

T11-1:アドレス表示 1

(本提出期限 2008/6/26(木)17:40, 再提出期限 2008/7/10(木)14:30)

提出物:Makefile, ソースファイル:print_address1.c, 入力ファイル:print_address1.in,
出力ファイル:print_address1.out

T10-1 の提出したプログラム (mymath.c) を, 宣言したローカル変数, 仮引数のアドレスを表示するように修正したプログラムを作成せよ. 次の実行例を参考に後ろの要求仕様を満たすこと.

実行例

```
b09b0xx@tyy:~/T11/1$ ./print_address1<print_address1.in
関数 main 中の変数 x のアドレス:0xbf90e6c8
関数 main 中の変数 y のアドレス:0xbf90e6c0
x の平方根を求めます.
x=?関数 mysqrt の仮引数 x のアドレス:0xbf90e670, 値:2.00000000
関数 mysqrt 中の変数 old_x のアドレス:0xbf90e688
関数 mysqrt 中の変数 new_x のアドレス:0xbf90e680
関数 mysqrt 中の変数 sa のアドレス:0xbf90e678
関数 mysqrt 中の変数 kaisuu のアドレス:0xbf90e694
関数 myfabs の仮引数 x のアドレス:0xbf90e640, 値:-2.00000000
関数 myfabs の仮引数 x のアドレス:0xbf90e640, 値:-0.50000000
関数 myfabs の仮引数 x のアドレス:0xbf90e640, 値:-0.08333333
関数 myfabs の仮引数 x のアドレス:0xbf90e640, 値:-0.00245098
関数 myfabs の仮引数 x のアドレス:0xbf90e640, 値:-0.00000212
2.00 の平方根は, y=mysqrt(x)=1.41421356 です.
b09b0xx@tyy:~/T11/1$
```

全体的な仕様:

- 以下の項目を除いて、**T10-1** 全体的な仕様を満たす。
- `main` 関数以外でも、アドレスの表示のために標準出力を用いても良い。
- プログラム全体の入力、`main` 関数内で標準入力から 0 以上の実数値を読み込む。
- プログラム全体の出力は、以下を行なう。
 - － すべての出力は標準出力に行なう。
 - － `main` 関数を含めたすべてのローカル変数のアドレスを出力する
 - － `main` 関数以外の関数では、各関数が呼び出された回数だけ次を出力する。
 - * 仮引数のアドレスとその仮引数に代入されている値 (実引数)
 - * 関数内で宣言したローカル変数のアドレス
 - － アドレスの出力は、変数宣言の直後に行なうこと。

絶対値を求める関数 `myfabs` の仕様

- **T10-1** の `myfabs` の仕様と同様とする。
- 仮引数 `x` に関して、アドレスと呼び出されたときに変数 `x` に格納されている値を標準出力へ出力する。
- 宣言した変数について、宣言直後にアドレスを標準出力へ出力する。

平方根を求める関数 `mysqrt` の仕様

- **T10-1** の `mysqrt` の仕様と同様とする。
- 仮引数 `x` に関して、アドレスと呼び出されたときに変数 `x` に格納されている値を標準出力へ出力する。
- 宣言した変数について、宣言直後にアドレスを標準出力へ出力する。

応用問題

T11-2:アドレス表示 2

(本提出期限 2008/7/3(木)17:40, 再提出期限 2008/7/10(木)14:30)

提出物:Makefile, ソースファイル:print_adress2.c, 入力ファイル:print_address2.in,

出力ファイル:print_address2.out

プログラム内の各変数について,

- 変数のアドレス
- 変数に格納されている値

プログラム内の各ポインタについて,

- ポインタのアドレス
- ポインタに格納されている値
- ポインタが指す変数に格納されている値

を表示するプログラムを作成せよ. ただし, 次の実行例を参考に, 後ろの要求仕様を満たすように作成せよ.

実行例

```
b09b0xx@tyy:~/T11/2$ ./print_address2
i_data1?
1
i_data2?
2
d_data1?
3.0
d_data2?
4.0
main 関数内での表示
function 呼出前
&i_data1=0xbfb0609c i_data1=      1
&i_data2=0xbfb06098 i_data2=      2
&i_ptr  =0xbfb06094 i_ptr  =0xbfb06098 *i_ptr  =      2
&d_data1=0xbfb06088 d_data1=  3.000000
&d_data2=0xbfb06080 d_data2=      0
&d_ptr  =0xbfb06090 d_ptr  =0xbfb06080 *d_ptr  =  4.000000

function 関数内での表示
&i_data1=0xbfb06070 i_data1=      1
&i_data2=0xbfb06064 i_data2=-1209028620
&i_ptr  =0xbfb06060 i_ptr  =0xbfb06064 *i_ptr  ==-1209028620
&d_data1=0xbfb06040 d_data1=  3.000000
&d_data2=0xbfb06050 d_data2=-1209991837
&d_ptr  =0xbfb0605c d_ptr  =0xbfb06050 *d_ptr  = -0.000000

main 関数内での表示
function 呼出後
&i_data1=0xbfb0609c i_data1=      1
&i_data2=0xbfb06098 i_data2=      2
&i_ptr  =0xbfb06094 i_ptr  =0xbfb06098 *i_ptr  =      2
&d_data1=0xbfb06088 d_data1=  3.000000
&d_data2=0xbfb06080 d_data2=      0
&d_ptr  =0xbfb06090 d_ptr  =0xbfb06080 *d_ptr  =  4.000000
b09b0xx@tyy:~/T11/2$
```

要求仕様 (T11-2)

全体的な仕様:

- ローカル変数のアドレスと値を表示する関数 `function` を作成して利用する。関数 `function` の仕様は後ろ参照。
- `main` 関数内のローカル変数として、`int` 型の変数を 2 個、`double` 型の変数を 2 個、`int *`型のポインタを 1 個、`double *`型のポインタを 1 個を用意すること。
- `int` 型の 1 つの変数に蓄えられている値は関数 `function` の実引数とすること。また、もう一つの `int` 型の 1 つの変数のアドレスは、`int *`型のポインタへ代入すること。
- `double` 型の 1 つの変数に蓄えられている値は関数 `function` の実引数とすること。また、もう一つの `double` 型の 1 つの変数のアドレスは、`double *`型のポインタへ代入すること。
- `main` 関数以外では、標準入出力は用いないこと。
- プログラム全体の入力、`main` 関数内で標準入力から読み込む。整数値を 2 個順に読み込み、その後で実数値を 2 個順に読み込むこと。
- 関数 `function` を 1 度呼び出すこと。
- `main` 関数内での出力は次を満たすこと。
 - － 以下の項目を、関数 `function` 呼出前および呼出後の 2 回、標準出力へ適切に整形して出力する。
 - － すべての変数のアドレス及び値を出力する。
 - － すべてのポインタでは、上記に加えてポインタが指す変数の値を出力する。

ローカル変数を表示する関数 `function` の仕様

- 次のプロトタイプ宣言を持つ.

```
/*
説明：
ローカル変数を表示する関数

引数：
i_data1:int 型の値を保持
d_data1:double 型の値を保持

戻り値：
なし (void)

副作用：
仮引数, ローカル変数のアドレスと値を標準出力へ出力する.
*/
void function(int i_data1,double d_data1);
```

- 一番目の仮引数 `i_data1` は, 任意の整数 (int 値) とする. 二番目の仮引数 `d_data1` は, 任意の整数 (double 値) とする.
- 戻り値はないものとする. (戻り値および関数の型は `void` とする.)
- 関数 `function` 内において, `main` 関数と (仮引数以外の) 同じ名前のローカル変数を用意すること.