

# 問題1 解答例 変数のアドレスの表示

int型変数xを初期化せずに定義し、そのアドレスとメモリ内の値を表示するプログラムを作成せよ.

実行結果例:

アドレスは0x7fff5fbffadcです.

値は0です.

```
#include <stdio.h>
int main(void)
{
    int x;
    printf("アドレスは%pです. ¥n値は%dです. ¥n", &x, x);
    return 0;
}
```

## 問題2解答例 配列のアドレスの表示

double型配列x[3]を初期化せずに定義し、配列の各要素のアドレスとメモリ内の値を表示するプログラムを作成せよ。

また、ポインタを用いて全ての要素に1.23を代入せよ。

```
#include <stdio.h>
```

```
int main(void)
```

```
{    int i;
```

```
    double x[3];
```

```
    for(i = 0; i < 3; i++)
```

```
    {    printf("x[%d]のアドレスは%pです.  \n  
        値は%fです.  \n", i, &x[i], x[i]);
```

```
        *(x+i) = 1.23;
```

```
    }
```

```
    return 0;
```

```
}
```

実行結果例:

x[0]のアドレスは0x7fff5fbffac0です.

値は0.000000です.

x[1]のアドレスは0x7fff5fbffac8です.

値は0.000000です.

x[2]のアドレスは0x7fff5fbffad0です.

値は0.000000です.

### 問題3 解答例

問題：左のプログラムを完成させるために  
入れるべき適当な文字や記号を答えよ。

```
main()
{ int num1=1, num2=2;
  int *ptr1 = &num1, *ptr2 = &num2;
  /*ポインタ変数ptr1の初期値として, num1の先頭アドレスを代入*/

  printf("num1 = %d\n", *ptr1);
  /*ポインタ変数ptr1が指し示すアドレスを先頭とするメモリ領域に
    保存された変数(num1)の値を画面に表示*/

  printf("num2 = %d\n", *ptr2);
  ptr1 = ptr2; もしくは ptr1 = &num2;
  /*ptr1の値をnum2 の先頭アドレスに変更*/

  *ptr1 = 5; /*ptr1を利用してnum2 の値を間接的に5に変更*/
  printf("num1 = %d\n", num1);
  printf("num2 = %d\n", num2);
}
```

実行結果: ./a.out

num1 = 1

num2 = 2

num1 = 1

num2 = 5

## 問題4 解答例

問題: int型の2つの変数の値を, それぞれを指すポインタを用いて交換するプログラムを作成し, プログラムと実行結果を示しなさい.

```
#include <stdio.h>
int main()
{   int x=1, y=23, temp;
    int *ptr1 = &x, *ptr2 = &y;
    printf("x=%d\t y=%d\n", x, y);
    temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
    printf("x=%d\t y=%d\n", x, y);
}
```

```
$ ./a.out
x=1    y=23
x=23   y=1
```

## 問題5 解答例

問題: int型の変数と配列(要素数3)を宣言し, それぞれのアドレスを表示するプログラムを作成し, プログラムと実行結果を示しなさい.  
配列は要素番号ごとのアドレスを表示すること.

```
#include <stdio.h>
int main()
{   int x=0, ary[3], *ptr=ary;
    printf("&x=%p\n", &x);
    while(x < 3)
        printf("&ary[%d]=%p, ptr=%p\n",
                x, &ary[x++], ptr++);
    return 0;
}
```

問題ではptrの表示は不要

```
&x=0x7fff6aed9bf0
&ary[0]=0x7fff6aed9be4, ptr=0x7fff6aed9be4
&ary[1]=0x7fff6aed9be8, ptr=0x7fff6aed9be8
&ary[2]=0x7fff6aed9bec, ptr=0x7fff6aed9bec
```