

1. 次のコードの出力を解答欄に答えよ。次に①の x , y にキーボードから任意の数値を入力出来るように `main()` 関数を変更し、変更点を解答欄に書け。

```
#include <stdio.h>

void func(int *a, int *b) {
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

int main() {
    int x = 3, y = 5;    . . . ①
    func(&x, &y);
    printf("%d %d\n", x, y);
    return 0;
}
```

2. 下記は配列要素を昇順に並び替えるプログラムである。「// ここを埋めよ」の部分を完成させてこの部分を解答欄に書け。

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

int main() {
    int n, i, j, tmp;
    scanf("%d", &n);
    int *arr = (int *)malloc(sizeof(int) * n);
    for (i = 0; i < n; i++) scanf("%d", &arr[i]);

    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // ここを埋めよ
            }
        }
    }

    for (i = 0; i < n; i++) printf("%d ", arr[i]);
    free(arr);
    return 0;
}
```

3. 構造体とは、複数の異なるデータ型を1つのまとまりとして定義できる仕組みである。

(例) 下記の定義で、name, age, height というデータを持つ Person という名の構造体が出来る。

```
typedef struct Person {  
    char name[50];  
    int age;  
    float height;  
} Person;
```

次の構造体 Node を使って、単方向リストの末尾に要素を追加する関数 `append()` を実装し、解答欄に書け。

単方向リストとは、データ構造の1種で、

- 1) 各要素 (ノード) がデータと次のノードへのポインタを持ち、
- 2) 一方向にしかたどれない、
- 3) 動的にサイズを変えられる、

という特徴を持っている。

(例) [データ 1 | 次へのポインタ] → [データ 2 | 次へのポインタ]
→ [データ 3 | NULL]

```
typedef struct Node {  
    int value;  
    struct Node *next;  
} Node;
```

```
void append(Node **head, int value);
```