

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

## 問題 1

以下の C 言語で書かれたプログラムについて考える。

```
#include <stdio.h>

int h(int[], int);
int start = 1;
int vec[8];

int main(void)
{
    int i;
    vec[0] = 3;
    vec[1] = 3;
    vec[2] = 6;
    vec[3] = 5;
    vec[4] = 4;
    vec[5] = 2;
    vec[6] = 0;
    vec[7] = 5;
    h(vec, start);
    return 0;
}

int h(int v[], int s)
{
    int i = s;
    while (v[i] != 0) {
        printf("-> %d\n", v[i]);
        i = v[i];
    }
    return i;
}
```

### 設問 1

大域変数 (グローバル変数) はどれか、名前を全て挙げよ。

解答欄：

### 設問 2

仮引数はどれか、名前を全て挙げよ。

解答欄：

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

### 設問 3

このプログラムを実行するとどのような出力が得られるか。正確に記せ。

解答欄：

### 設問 4

関数 main 中の変数 vec への代入文において代入する値をいくつか適当に変更して (ただし代入する値は 0 から 7 までのいずれかとする) 実行すると出力が無限に続き、止まらなくなった。そこで、代入後の vec を基に関数 h の出力が無限に続くかどうかを判定し、出力が止まらないなら 0 を、止まるなら 1 を返す関数 teishiHantei を考える。この関数は関数 h と同じように vec と start とを引数に取るものとして作成せよ。

解答欄：

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

## 問題 2

以下に配列変数に代入された整数値を昇順にソートするプログラムを示す。関数 `swap` は整数型配列、配列の要素番号 1 および要素番号 2 を引数として、配列の要素番号 1 と要素番号 2 の内容を入れ替える。関数 `simple_sort` は整数型配列および配列長を引数として配列中の整数値を昇順にソートする。このプログラムについて以下の設問に答えよ。

なお、以下の設問において必要な変数は適宜宣言して使用して良い。

### 設問 1

解答欄ア、イ、ウの空欄を埋めて、このプログラムを完成せよ。

```
#include <stdio.h>

//関数宣言

void swap(  );
void simple_sort(  );

void swap(int array[], int e1, int e2)
{
    
}

void simple_sort(int target[], int length)
{
    int i, j, k;
    int min; // 変数 min には探索した配列
            // データ中で最小値のある要素番号を入力する
    for(i=0; i < length - 1; i++){
        min = i;
        for(j=i+1; j < length; j++){
            if(target[j] < target[min]){
                min = j;
            }
        }
        swap(target,i,min);

        printf("%d :", i);
        for(k=0; k < length; k++)
            printf("%3d", target[k]);
        printf("\n");
    }
}

int main(void)
{
    int test[]={9, 2, 1, 5, 4, 3, 7, 6, 8, 10};

    simple_sort(test, 10);

    return 0;
}
```

解答欄：

ア	
イ	
ウ	

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

設問 2

設問 1 のプログラムを実行するとどのような出力が得られるか。以下の解答欄に記せ。

解答欄：

i の値	出力結果
0	
1	
2	
3	
4	
5	
6	
7	
8	

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

### 設問 3

設問 1 の関数 `simple_sort` より高速なソート関数 `new_sort` を、解答欄エおよびオの空欄を埋めて作成せよ。もし以下の解答欄に書ききれない場合には、その旨を明記してこの用紙の裏面にわたっても良い。なお、設問 1 の関数 `simple_sort` 中にあるような処理の経過を `printf` 関数で出力する部分は必要はない。

解答欄：

エ	
オ	

```
void new_sort(  )  
{  
      
}
```