課題 2a:配列を用いた整列挿入

それまでに探索された整数すべてを小さい順に並べた整数列を配列に格納しておき，ある整数値 key を引数として呼び出されたら，key が探索されたことがなければ 0(No の意味) を返すとともに引数を小さい順に並ぶように指数列配列に格納し，key が既に探索したことがあれば 1(yes の意味) を返すだけのという，関数 (int insert_sorted_list()) を作れ.

以下に今回作成した inset_sorted_list 関数とそれを構成する関数群を示す.

```
/*配列の要素数*/
#define SIZE 100
/*配列を要素数 SIZE で宣言，0 で初期化*/
static int ARRAY[SIZE] = {};

/*配列のどこまで要素が代入されたかを示す変数*/
int ARRAYEND = 0;
/**
 * @brief 配列の中身を 3 桁ごとに表示する関数
 * @return なし
 */
void print_array()
{
    int i;
    /*配列の中身全てではなく代入された要素だけを表示*/
    for (i = 0; i < ARRAYEND; ++i)
    {
        printf("%3d,", ARRAY[i]);
    }
    printf("\n");
}

/**
 * @brief 配列の要素を昇順にソートする関数
 * @return なし
 */
void asc_sort_array()
{
    int i, j, temp;
    for (i = 0; i < ARRAYEND; ++i)
    {
        for (j = i + 1; j < ARRAYEND; ++j)
```

```
        {
            /*i 番目の要素より小さい要素があったら入れ替える*/
            if (ARRAY[i] > ARRAY[j])
            {
                temp = ARRAY[i];
                ARRAY[i] = ARRAY[j];
                ARRAY[j] = temp;
            }
        }
    }
}

/**
 * @brief 配列の中に key と同値が存在するか探索し昇順にソートする.
 * @param key:配列に格納したい変数
 * @return 配列に既に key が存在すれば 1(Yes の意味)
 *          存在しなければ 0(No の意味)
 */
int insert_sorted_list(int key)
{
    printf("insert:%d\n",key);
    int i;
    /*配列の中に key と同じ要素がないか探索する*/
    for (i = 0; i < ARRAYEND; ++i)
    {
        if (ARRAY[i] == key)
        {
            /*存在した場合ソートだけ行う*/
            asc_sort_array();
            printf("Yes\n");
            print_array();
            return 1;
        }
    }
    /*存在しなかった場合，配列の後ろに key の値を代入*/
    ARRAY[ARRAYEND] = key;
    /*どこまで代入したかを示す ARRAYEND をインクリメント*/
    ARRAYEND++;
    /*ソートを実行*/
```

```
    asc_sort_array();
    printf("No\n");
    print_array();
    return 0;
}
```

メールに添付したソースコードを実行すると以下のような出力を得られる．

```
insert:8
No
  8,
insert:3
No
  3,  8,
insert:12
No
  3,  8, 12,
insert:1
No
  1,  3,  8, 12,
insert:8
Yes
  1,  3,  8, 12,
insert:10
No
  1,  3,  8, 10, 12,
insert:5
No
  1,  3,  5,  8, 10, 12,
insert:0
No
  0,  1,  3,  5,  8, 10, 12,
insert:3
Yes
  0,  1,  3,  5,  8, 10, 12,
```

今回作成した関数では，引数 key が与えられて insert_sorted_list が呼び出されたら，配列の中身を探索し，key と同じ要素が存在した場合，配列を昇順にソートする asc_sort_array 関数を呼びだし配列をソートした後，既に探索したことを示す Yes を表示し，配列の中身を表示する関数 print_array を呼びだし 1 を return している．一方，配列の中身に key と同じ要素がなかった場合，配列のまだ代入されていない位置に key の値を代入し，配列に代入された要素数を示すグローバル変数 ARRAYEND をインクリメン

トする．その後，ソートを行い，探索していないことを示す No を，表示し，配列の中身を表示して 0 を return している．

考察感想

配列を用いる場合では，配列を宣言し初期化したあと，要素を代入していったときに，どこまで代入したのかを判別する変数 (自分のプログラムだと ARRAYEND) を用いる必要があったので不便に感じた．また，ソートを毎回行っているのでソートされている配列とわかっている場合ならば，間に挿入できるリスト構造のほうが明らかに効率的であると考える．

今回のプログラムではソートの部分は重要ではないと考えたので最適化高速化等は行っていない．

協力者

19257504:太田那菜