

問題用紙

2026	科目名	生命化学情報：情報科学Ⅱ	1 / 3	通し番号	
------	-----	--------------	-------	------	--

問題 下の code_01.c は、DNA に関するデータ（DNA の名前と配列）を入力ファイルとして読み込み、解析を行った後、解析結果を出力ファイルに書き出す。下の dna_input.csv は入力ファイルの一例である。入力ファイルと出力ファイルはともにテキストファイルである。code_01.c に関する次の問いに答えよ。（各問 5 点）

問 1 空欄(1)に適切なコードを記せ。

問 2 空欄(2)に適切なコードを記せ。

問 3 空欄(3)に適切なコードを記せ。

問 4 空欄(4)に適切なコードを記せ。

問 5 空欄(5)に適切なコードを記せ。

問 6 空欄(6)に適切なコードを記せ。

問 7 関数 sort で採用されているソートアルゴリズムの計算量をオーダー記法で答えよ。

問 8 入力ファイルが多くのレコードを含む場合、関数 sort の計算量が非常に大きくなる。計算量を小さくするには関数 sort をどのように改良すべきか。簡潔に答えよ。

問 9 入力ファイル dna_input.csv を与えたときの出力ファイル dna_output1.csv の内容を記せ。

問 10 入力ファイル dna_input.csv を与えたときの出力ファイル dna_output2.csv の内容を記せ。

dna_input.csv

DNA-1,GGGC DNA-2,TATCATGA DNA-3,GGTCAA
--

```
/***** code_01.c *****/
```

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

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

```
#include <string.h>
```

```
typedef struct {
```

```
    char name[100];    // DNA の名前
```

```
    char seq[100];    // DNA の配列
```

```
    int len;          // DNA の長さ
```

```
    double gc_content; // GC 含量
```

```
} DNA;
```

(code_01.c は次ページに続く)

2026	科目名	生命化学情報：情報科学Ⅱ	2 / 3	通し番号	
------	-----	--------------	-------	------	--

```
double calc_gc_content(DNA *dna);
void sort(DNA records[], int n);
void swap(DNA *a, DNA *b);
void write_to_file(DNA records[], int n, char filename[]);

int main() {
    FILE *fp;
    char name[100], seq[100];
    DNA records[100];
    int n = 0;

    // ファイルを開く
    if ((fp = fopen("dna_input.csv", "r")) == NULL) {
        fprintf(stderr, "Failed to open input file.\n");
        exit(1);
    }

    // ファイルを読み込んで構造体に格納
    while (fscanf(fp, "%[^,],%s\n", name, seq) != EOF) {
        strcpy(records[n].name, name); // 名前をコピー
        strcpy(records[n].seq, seq); // 配列をコピー
        (1) = strlen(seq); // 配列の長さを計算
        records[n].gc_content = (2); // GC 含量を計算
        n++;
    }
    fclose(fp);

    write_to_file(records, n, "dna_output1.csv");

    sort(records, n);

    write_to_file(records, n, "dna_output2.csv");

    return(0);
}
```

(code_01.c は次ページに続く)

2026	科目名	生命化学情報：情報科学Ⅱ	3 / 3	通し番号	
------	-----	--------------	-------	------	--

// GC 含量を計算。GC 含量は、DNA 配列中の G と C が占める割合を百分率で表したもの

```
double calc_gc_content(DNA *dna) {
    int i, gc = 0;
    double gc_content = 0.0;

    for (i = 0; i < dna->len; i++)
        if (  )
            gc++;

    if (dna->len > 0)
        gc_content =  * 100.0;

    return(gc_content);
}

// DNA をソート
void sort(DNA records[], int n) {
    int i, j;
    for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - i - 1; j++)
            if (records[j].gc_content > records[j + 1].gc_content)
                ;
}

// 構造体を入れ替え
void swap(DNA *a, DNA *b) {
    DNA temp = *a;
    *a = *b;
    *b = temp;
}

// ファイル出力関数
void write_to_file(DNA records[], int n, char filename[]) {
    int i;
    FILE *fp;

    if (  ) {
        fprintf(stderr, "Failed to open output file \"%s\".\n", filename);
        exit(1);
    }

    for (i = 0; i < n; i++)
        fprintf(fp, "%s,%s,%d,%.2f\n", records[i].name, records[i].seq,
records[i].len, records[i].gc_content);

    fclose(fp);
    printf("Data has been written to \"%s\".\n", filename);
}
}
```