

受験番号	解答例
------	-----

令和7年度
新潟大学工学部第3年次編入学
学 力 試 験

解答は各問とも必ずこの試験用紙に記入すること

試 験 科 目	専門基礎科目 (プログラミング)	1 / 2 頁
---------	------------------	---------

〔I〕以下の問に答えよ。解答は各問の下に記入すること。

- (1) m 個 ($m \geq 1$) の互いに区別のできないボールを、 n 個 ($n \geq 1$) の互いに区別のできない袋すべてに詰める仕方の数を $p(m,n)$ と表す。例えば、5個のボールを3つの袋に詰めるには、 $(3,1,1)$ 、 $(2,2,1)$ の2通りなので、 $p(5,3) = 2$ である。どんな数 m に対しても、 $p(m,1) = 1$ 、 $p(m,m) = 1$ は自明である。袋すべてに詰めねばならないので、 $m < n$ に対しては、 $p(m,n) = 0$ である。さて、 $m > n$ に対して、

$$p(m,n) = p(m-n,n) + p(m-n,n-1) + \cdots + p(m-n,2) + p(m-n,1)$$

が与えられることが分かった。このとき、以下の問に答えよ。

- ①関数 $p(m,n)$ が既に作成されているとき、キーボードから m と n の値を入力して、 $p(m,n)$ の値を出力する main 関数を C 言語で書け。使用した変数やプログラムの動作に関する部分には可能な限りコメントをつけること。

```
/* メイン関数 */
int main(void) {
    int m, n; /* m と n を整数型として定義 */

    scanf("%d", &m); /* m の入力 */
    scanf("%d", &n); /* n の入力 */

    printf("p(%d, %d) = %d\n", m, n, p(m, n)); /* p(m,n) を出力 */

    return 0; /* main 関数終了 */
}
```

- ② $p(m,n)$ を計算する関数を C 言語で書け。ただし、再帰呼び出しを用いること。使用した変数やプログラムの動作に関する部分には可能な限りコメントをつけること。

```
/* p(m,n) を計算する関数 */
int p(int m, int n) {
    int i, p_sum = 0;

    if (n == 1 || n == m) /* p(m,1) と p(m,m) は 1 */
        return 1;

    if (m < n) /* m < n に対しては p(m,n)=0 */
        return 0;

    /* p(m-n,i) を i=1 から n まで足し合わせたものが p(m,n) となる。再帰を利用 */
    for (i = 1; i <= n; i++)
        p_sum += p(m - n, i);

    return p_sum; /* 上記で足し合わせたものを返す */
}
```

受験番号	解答例
------	-----

令和7年度

新潟大学工学部第3年次編入学

学 力 試 験

解答は各問とも必ずこの試験用紙に記入すること

試 験 科 目	専門基礎科目 (プログラミング)	2 / 2 頁
---------	------------------	---------

[II] 次のプログラムは認識した英数字列を順番にリストに格納し、最後に出力するものである。このプログラムを参考にして、あらたに同一の英数字列の出現数をカウントして英数字列とその出現数を出力するプログラムを作成したい。このとき英数字列の出現順序は考慮しない。

- (1) このプログラムのどの部分をどのように変更すれば実現できるか。できる限り効率よく実現できる方法を説明せよ。
- (2) (1) で説明した方法を実現するコードを示せ。変更部分のみを示すだけで良い。このとき英数字列の辞書式順序での大小比較を行う関数 strcmp は使って良い。

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct worddata {
6      struct worddata *next;
7      char *str;
8  } worddata;
9  worddata *wordList = NULL;
10
11 int printList()
12 {
13     worddata *car = wordList;
14     while (car != NULL) {
15         printf("%s\n", car->str);
16         car = car->next;
17     }
18     return 0;
19 }
20
21 int main()
22 {
23     int c;
24     char buf[256];
25     int i = 0;
26     while ((c = getchar()) != EOF) {
27         if ( ('a' <= c && c <= 'z') ||
28             ('A' <= c && c <= 'Z') ||
29             ('0' <= c && c <= '9') ) {
30             buf[i++] = c;
31         } else if (i > 0) {
32             worddata *newelem;
33             newelem = malloc(sizeof(worddata));
34             newelem->next = NULL;
35             newelem->str = malloc(i+1);
36             strncpy(newelem->str, buf, i);
37             if (wordList==NULL) wordList = newelem;
38             else {
39                 worddata *car = wordList;
40                 while (car->next != NULL)
41                     car = car->next;
42                 car->next = newelem;
43             }
44             i = 0;
45         }
46     }
47     printList();
48 }

```

(1)

構造体 worddata に出現数をカウントする変数 count を追加する。認識した英数字列をリストに格納する時に、線形リストになるように格納位置を決定する。このとき、同一英数字列が見つかれば count をインクリメントし、見つからない場合は決定した位置に挿入する。

(2)

5行 struct worddata に次を追加

int count;

13行 printList の printf を次に変更

printf("%s, %d\n", car->str, car->count);

37行から43行を次のように変更

(裏面に続く)

```

newelem->count = 1;
if (wordList==NULL) wordList = newelem;
else {
    worddata *car = wordList;
    if (car->next == NULL) {
        if (strcmp(newelem->str, car->str) == 0) {
            car->count++;
        } else if (strcmp(newelem->str, car->str) < 0) {
            wordList = newelem;
            newelem->next = car;
        } else {
            car->next = newelem;
        }
    } else {
        if (strcmp(newelem->str, car->str) == 0) {
            car->count++; free(newelem->str); free(newelem);
        } else if (strcmp(newelem->str, car->str) < 0) {
            wordList = newelem;
            newelem->next = car;
        } else {
            while (car->next != NULL)
                if (strcmp(newelem->str, car->next->str) > 0)
                    car = car->next;
                else
                    break;
            if (car->next != NULL && strcmp(newelem->str, car->next->str) == 0)
                car->next->count++;
            else {
                newelem->next = car->next;
                car->next = newelem;
            }
        }
    }
}
}
}
}

```