

次の問9から問13までの5問については、この中から1問を選択し、答案用紙の選択欄の(選)をマークして解答してください。
なお、2問以上選択した場合には、はじめの1問について採点します。

問9 次のCプログラムの説明及びプログラムを読んで、設問1、2に答えよ。

(プログラムの説明)

C言語では整数型の変数に格納できる値には上限がある。これを超える正の整数(以下、正の多倍長整数という)の入出力と加算を行うプログラムである。

- (1) 正の多倍長整数は、次に示すMP型の構造体を用いて表現する。

```
typedef struct{
    int length;
    long data[ARRAY_MAX];
}MP;
```

正の多倍長整数を、下位から9けたずつに切り分けて、構造体MPのメンバである配列dataの要素番号の小さい方から順番に値を格納する。例えば、46284059827463859201283844157134007652918723147641という整数の場合、図のとおりになる。

46284|059827463|859201283|844157134|007652918|723147641

要素番号	5	4	3	2	1	0
配列data	46284	59827463	859201283	844157134	007652918	723147641

図 正の多倍長整数の格納例

構造体のメンバlengthには、実際に値を格納した要素数を入れる。図の場合は6である。

(2) 関数の仕様は、次のとおりである。

```
void set(MP *num, const char str[]);
```

引数 : num MP型の構造体で表現された多倍長整数

str 文字列で表現された多倍長整数であって、1～9の数字で始まる。

機能 : 文字列で与えられた多倍長整数 str を変換して、MP型の構造体 num に格納する。num のメンバ data は、変換後の数値を格納するのに十分な要素数が確保されているものとする。

返却値 : なし

```
void print(const MP *num);
```

引数 : num MP型の構造体で表現された多倍長整数

機能 : 多倍長整数を出力する。

返却値 : なし

```
void add(const MP *a, const MP *b, MP *c);
```

引数 : a, b, c MP型の構造体で表現された多倍長整数であり、c は、a 及び b ではないものとする。

機能 : 二つの多倍長整数 a, b の和を多倍長整数 c に格納する。c のメンバ data は、加算処理を行うのに十分な要素数が確保されているものとする。

返却値 : なし

(3) 次のライブラリ関数を用いる。

```
size_t strlen(const char *s);
```

機能 : s が指す文字列の長さを計算する。

返却値 : 終端を示すナル文字に先行する文字の個数を返す。

(プログラム)

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

#define ARRAY_MAX 100
#define NUM_DIGIT 9
#define NUM_DIGIT_TH_POWER_OF_TEN 10000000000

typedef struct{
    int length;
    long data[ARRAY_MAX];
}MP;

void set(MP*, const char[]);
void print(const MP*);
void add(const MP*, const MP*, MP*);

/* 文字列から多倍長整数を扱う構造体に変換 */
void set(MP *num, const char str[]){
    int str_idx = strlen(str) - 1;
    int num_idx = 0;
    int i;
    long mul;

    while( [a] ){
        num->data[num_idx] = 0;
        mul = 1;
        for(i = 0; [b]; i++){
            num->data[num_idx] += mul * (str[str_idx--] - '0');
            mul [c];
        }
        num_idx++;
    }
    num->length = num_idx;
}

/* 多倍長整数の出力 */
void print(const MP *num){
    int i;

    printf("%ld", num->data[num->length - 1]);
    for([d]){
        /* ゼロ詰めして必ず9けたを表示する。 */
        printf("%09ld", num->data[i]);
    }
    printf("\n");
}
```

```

/* 二つの多倍長整数の加算 */
void add(const MP *a, const MP *b, MP *c){
    int i;
    int i_max;

    if(a->length > b->length){
        i_max = a->length;
    }else{
        i_max = b->length;
    }
    c->data[0] = 0;

    for(i = 0; i < i_max; i++){
        if(i < a->length) c->data[i] += a->data[i];
        if(i < b->length) c->data[i] += b->data[i];

        if(c->data[i] >= NUM_DIGIT_TH_POWER_OF_TEN){  

            c->data[i + 1] = 1;  

            c->data[i] -= NUM_DIGIT_TH_POWER_OF_TEN;  

        }else{  

            c->data[i + 1] = 0;  

        }
    }
    if(c->data[i] == 0){  

        c->length = i;  

    }else{  

        c->length = i + 1;  

    }
}

```

α

β

設問1 プログラム中の [] に入る正しい答えを、解答群の中から選べ。

aに関する解答群

- | | | |
|----------------|----------------|----------------|
| ア str_idx != 0 | イ str_idx < 0 | ウ str_idx <= 0 |
| エ str_idx > 0 | オ str_idx >= 0 | |

bに関する解答群

- | | |
|---------------------------------|--|
| ア i < NUM_DIGIT | |
| イ i < NUM_DIGIT && str_idx <= 0 | |
| ウ i < NUM_DIGIT && str_idx >= 0 | |
| エ i < NUM_DIGIT str_idx <= 0 | |
| オ i < NUM_DIGIT str_idx >= 0 | |

cに関する解答群

ア % = 10 イ *= 10 ウ += 10 エ -= 10 オ /= 10

dに関する解答群

ア i = 0; i <= num->length - 1; i++
イ i = 0; i <= num->length - 2; i++
ウ i = num->length - 1; i >= 0; i--
エ i = num->length - 2; i >= 0; i--

設問2 プログラムの動作について、次の記述中の [] に入る正しい答えを、
解答群の中から選べ。

関数 add を用いて以下の(1)に示す二つの多倍長整数を加算する場合、プログラム中の α 部分は [e] 回実行され、 β 部分は [f] 回実行される。
また、以下の(2)に示す二つの多倍長整数を加算する場合、 β 部分は [g] 回実行される。

(1) 231456813|123456789|234567890|478617283

999999999|123456789|009283741

(2) 85923|726438423|734239982|017238764|500000000|321987472

5472937|726325476|292278313|499999999|783917811

e～gに関する解答群

ア 0	イ 1	ウ 2	エ 3
オ 4	カ 5	キ 6	ク 7