

錬成問題

- 以下に示すのは、 x の二乗と三乗を求める関数形式マクロである。

```
#define sqr(x) ( (1) )
#define cub(x) ( (2) )
```

- 以下に示すのは、 x の絶対値を求める関数形式マクロである。

```
#define abs(x) ((x) > 0) ? (3) : (4)
```

- 以下に示すのは、二つの値 x , y の大きい方の値と小さい方の値を求める関数形式マクロである。

```
#define maxof(x, y) ((x) > (y) ? (5) : (6))
#define minof(x, y) ((x) < (y) ? (7) : (8))
```

- 以下に示すのは、二つの値 x , y の差を求める関数形式マクロである。

```
#define diff(x, y) ((x) > (y) ? (9) : (10))
```

- 以下に示すのは、`type`型の二つの値を交換する関数形式マクロである。

```
#define swap(type, x, y) do { (11) } while (0)
```

- 以下に示す(12)体の宣言が与えられているとする。

```
enum aaa { a1, a2 = 5, a3 };
```

ここで、`aaa`は(13)と呼ばれ、“`enum aaa`”が型となる。

また、`a1`, `a2`, `a3`は(14)と呼ばれ、それぞれの値は(15), (16), (17)である。

- 右に示す関数を`func(3)`と呼び出したときの実行結果を示せ。

```
(18)
```

```
void func(int no)
{
    if (no > 0) {
        func(no - 1);
        printf("%d ", no);
        func(no - 2);
    }
}
```

- 右に示すプログラム部分の実行結果を示せ。

```
(19)
```

```
#include <stdio.h>

printf("%d", EOF < 0);
```

■ 右に示すのは、 n の階乗値を返す関数である。このように、関数の中で自分と同じ関数を呼び出すことを、関数呼出しという。

```
int fact(int n)
{
    if (n > 0)
        return (n * );
    else
        return (1);
}
```

■ 以下に示すのは、123123... を no 文字だけ表示するプログラム部分である。

※ たとえば、 no が 8 であれば 12312312 と、4 であれば 1231 と表示する。

```
for (i = 0; i < no; i++)
    putchar();
```

■ 以下に示すのは、標準入力から次々と1文字ずつ文字を読み込み、それをそのまま標準出力へと書き出すプログラムである。

```
#include <stdio.h>

int main(void)
{
    int ch;

    while ((ch = ) != )
        putchar(ch);

    return (0);
}
```

■ 以下に示すのは、標準入力から次々と1文字ずつ文字を読み込み、読み込んだ文字数と行数を表示するプログラムである。

```
#include <stdio.h>

int main(void)
{
    int ch;
    int c_cnt = 0; /* 文字数 */
    int l_cnt = 0; /* 行数 */

    while ((ch = ) != ) {
        ++;
        if (ch == )
            ++;
    }
    printf("文字数: %d\n", c_cnt);
    printf("行数: %d\n", l_cnt);

    return (0);
}
```