

6つの設問の中から評価の高い5問の合計点を用いる。設問中の'¥'は'\ 'で示す。

## 1 用語穴埋め問題

以下の説明文の下線部に、ふさわしい用語を埋めるか、2つの選択肢の正しい1つを で囲め。

1. C言語のソースから機械語を作るには、最初に (A3)\_\_\_\_\_ によって#が含まれないソースに変換する。次にコンパイラによって (B3)\_\_\_\_\_に変換する。最後に、(C3)\_\_\_\_\_によって標準関数のライブラリと結合して実行ファイルが作られる。
2. #define マクロは、処理を登録しておいて再利用するという点では、通常関数と同じであるが、機械語を生成する前に (A) により文字列として置換される。このため、#define マクロを複数利用すると、マクロの内容の処理が複数作られ生成される機械語が (D2)短く/長くなる。一方で、(E3)\_\_\_\_\_の代入などが無いため処理速度は少し速くなる。
3. #define マクロは、(F3)閉じた/開いた関数とも呼ばれる。前述の理由から、処理の中身が (G3)短い/長い場合に、#define マクロを利用する。

## 2 プログラム穴埋め問題

以下に示すような3つのファイルに、名前と点数が記録されている。全データの平均点を求めるプログラムとなるように下線部を埋めよ。(小数点以下も正しく求まること)

```

void main() {
    char filename[ 20 ] ;
    int count = 0 , sum = 0 , i ;
    for( i = 0 ; i < 3 ; i++ ) {
        fp ;
        ~~~~~(A)2
        sprintf( filename , "file%d.dat" , i ) ;
        if ( (fp = ~~~~~) ) {
            char name[ 20 ] ;
            int point ;
            if ( ~~~~~ ) {
                count++ ;
                sum += point ;
            }
        } ~~~~~(E)4
    }
    printf( "average %lf\n" , ~~~~~ ) ;
}

```

| ((ファイル))  
 | file0.dat tohru 90  
 | file1.dat mitsuki 83  
 | file2.dat ayuka 80

(B)4 (C)2 (D)4 (F)4

### 3 相対 PATH, 絶対 PATH

Windows 環境で、ディレクトリの内容を表示する命令を `dir`, ファイルの内容を表示する命令を `type`, ディレクトリを指定する命令を `cd` とする。右図に示すようなディレクトリ構造が与えられ、

```
dir \aa\aa
```

を実行したら、`a, b` が表示される。  
この時、以下の設問に答えよ。

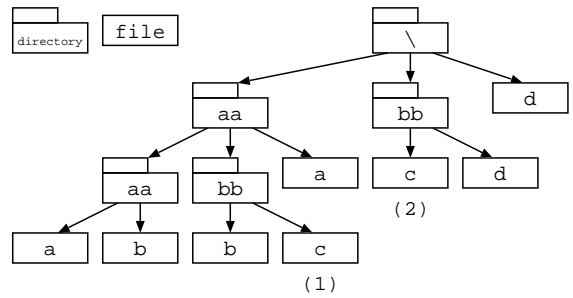
正しく実行できるコマンドは、エラーが表示されるコマンドは `x` を (A)-(D) の欄に記入せよ。(3x4)

```
z:\> dir \aa\aa
a b
z:\> cd \aa\bb
z:\aa\bb> dir \bb\aa (A)_____

z:\aa\bb> dir ..\..\bb (B)_____

z:\aa\bb> type .\a (C)_____

z:\aa\bb> type ..\a (D)_____
```



(1), (2) のファイルを表示するコマンドを、指定した PATH の記述方法で、記載せよ。(4x2)

(1) 絶対 PATH 表記

(2) 相対 PATH 表記 (`cd \aa\bb` 実行後)

### 4 基本プログラム作成問題

Windows にて、ファイル `z:\addr\pokestop.txt` に、場所の名前と緯度経度が 1 行 1 件で記録されている。この中から、関数引数の緯度 (`lat`), 経度 (`lon`) と誤差が  $\pm 0.001$  度 (例 135.0 度なら、134.999-135.001, 緯度も同様とする) 以内の場所の情報をファイル `z:\addr\area.txt` に出力する関数 `in_area()` を作成せよ。(20)

```
((z:\addr\pokestop.txt))
30th-monument 35.936481 136.171695
clock-tower 35.936690 136.172720
fukui-college 35.936894 136.170531
((呼出側))
void main() {
    in_area( 35.936916 , 136.172854 ) ;
    // "clock-tower 35.936690 136.172720"を記録
}
```

## 5 説明問題

1. Windows と Unix で行末文字の違いから、C 言語ではどのような対応がとられているか、具体的な例を交えて説明せよ。(10)
2. バッファオーバーフローとその対策について、具体的な例を交えて説明せよ。(10)

## 6 応用問題

ファイル `monster.dat` に、ポケモン名と戦闘力 (CP)

が 1 行 1 件で登録されている。このデータの中で、同じ名前のポケモンがいた場合、戦闘力の最も高いものを残し、戦闘力の低いデータを `printf()` で出力する

プログラムを作成せよ。(20)

ポケモン名は 16 文字以内とし、名前と戦闘力を保存する配列の大きさは、最大 10 件とする。

文字列関数: `strcmp(a,b)` は、`a,b` が同じ綴りの場合 0 を返す。`strcpy(a,b)` は、`b` を `a` にコピーする。

|                         |                 |  |                              |
|-------------------------|-----------------|--|------------------------------|
| <code>pikachu</code>    | <code>54</code> |  | この例では、                       |
| <code>jigglypuff</code> | <code>32</code> |  | <code>"pikachu 54",</code>   |
| <code>pikachu</code>    | <code>62</code> |  | <code>"jigglypuff 16"</code> |
| <code>snorlax</code>    | <code>19</code> |  | を出力したい。                      |
| <code>jigglypuff</code> | <code>16</code> |  |                              |