情報処理教育シリーズ

追補版

PASSPORT

パスポート

情報処理試験合格へのパスポート

# 基本情報・午後対策演習問題集

## C言語

We Net ウイネット

# 基本情報･午後対策演習問題集
# C言語　追補版

## 目　　次



問題に記載されている会社名又は製品名は，それぞれ各社の商標又は登録商標です。
なお，問題では，® 及び ™ を明記していません。

# C 言語

**問 1** 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〕

ASCII コードの半角文字とシフト JIS コードの全角文字が混在するファイルを標準入力から読み込み，全角文字部分を JIS 漢字コード(JIS X0208)に変換して，標準出力に書き出すプログラムである。

(1) シフト JIS コードは 2 バイト(16 ビット)を使用した文字コードであり，JIS の 1 バイト文字セット(JIS X0201, 半角文字を定義)で未定義領域となっている 8 ビットコードを，2 バイトのうちの 1 バイト目(上位 8 ビット)に割り当てたもので，次の範囲を使用している。

1 バイト目：0x81～0x9F 及び 0xE0～0xEF

2 バイト目：0x40～0x7E 及び 0x80～0xFC

(2) シフト JIS コードの全角文字を，JIS 漢字コードに変換するには，次のアルゴリズムを用いる。

① 1 バイト目が 0xE0 以上であれば，1 バイト目から 0x40 を引く。
② 1 バイト目から 0x81 を引き，1 ビット左へシフトする。
③ 2 バイト目が 0x80 以上であれば，2 バイト目から 1 を引く。
④ 2 バイト目が 0x9E 以上であれば，1 バイト目に 1 を加算し，2 バイト目から 0x9E を引く。そうでなければ，2 バイト目から 0x40 を引く。
⑤ 1 バイト目と 2 バイト目に，それぞれ 0x21 を加算する。

(3) 上記の変換を行うことで，全角文字（JIS 漢字コード）の 1 バイト目と 2 バイト目は，ともに 0x21～0x7E の範囲となる。しかし，このままでは半角文字との区別がつかなくなるため，全角文字の開始時と終了時に，次のような特別な符号 0x1B を付加する。

全角文字開始時（漢字 IN）　：0x1B 0x24('$') 0x40('@')

全角文字終了時（漢字 OUT）　：0x1B 0x28('(') 0x4A('J')

〔プログラム〕

```
#define sjis_b1(c) ((c)>=0x81 && (c)<=0x9F || (c)>=0xE0 && (c)<=0xEF)
#define sjis_b2(c) (      a      )
#define ESC 0x1B

void sjis_jis(int *b1, int *b2)  /* シフト JIS コードを JIS コードに変換 */
{
    if (*b1 >= 0xE0)  *b1 -= 0x40;
    *b1 =       b       ;
    if (*b2 >= 0x80)  *b2 -= 1;
    if (*b2 >= 0x9E) {
```

```
            *b1 += 1;  *b2 -= 0x9E;
        } else
            *b2 -= 0x40;
        *b1 += 0x21;  *b2 += 0x21;
    }

    #include <stdio.h>
    #include <stdlib.h>
    int main()
    {
        int   c, d, flg;

        [ c ];
        while ((c = getchar()) != EOF) {
            if (sjis_b1(c)) {
                d = getchar();
                if (sjis_b2(d)) {
                    if (flg) {
                        putchar(ESC); putchar('$'); putchar('@');
                        flg = 0;
                    }
                    [ d ];
                    putchar(c);  putchar(d);
                } else {
                    if (!flg) {
                        putchar(ESC); putchar('('); putchar('J');
                        flg = 1;
                    }
                    putchar(c);
                    if (d != EOF) putchar(d);
                }
            } else {
                if (!flg) {
                    putchar(ESC); putchar('('); putchar('J');
                    flg = 1;
                }
                putchar(c);
            }
        }
        return 0;
    }
```

**設問１**　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

a に関する解答群

ア　(c)>0x39 && (c)<=0x7E || (c)>0x79 && (c)<=0xFC

イ　(c)<0x40 || (c)>0x7E && (c)<0x80 || (c)>0xFC

ウ　!((c)<0x40 && (c)>0x7E || (c)<0x80 && (c)>0xFC)

エ　(c)>=0x40 && (c)<=0xFC && (c)!=0x7F

オ　(c)>=0x40 && (c)<=0xFC || (c)!=0x7F

b に関する解答群

ア　*b1 - 0x81 << 1　　イ　*b1 - 0x81 >> 1

ウ　*b1 << 1 - 0x81　　エ　*b1 >> 1 - 0x81

オ　*b1 * 2 - 0x81

c に関する解答群

ア　c = 0　　イ　c = EOF

ウ　d = EOF　　エ　flg = 0

オ　flg = 1

d に関する解答群

ア　sjis_jis(c, d)

イ　sjis_jis(*c, *d)

ウ　sjis_jis(&c, &d)

エ　sjis_jis(**c, **d)

オ　sjis_jis(sjis_b1(c), sjis_b2(d))

**設問２**　シフト JIS コードの「あ」は，0x82A0 である。これを JIS 漢字コードに変換したときのコードとして正しい答えを，解答群の中から選び，解答欄 e にマークせよ。

解答群

ア　0x2321　　イ　0x2341　　ウ　0x2421　　エ　0x2422

# C 言語　解答･解説

＜解答＞

| | a | b | c | d | e | f | g | h | i | j |
|---|---|---|---|---|---|---|---|---|---|---|
| 問１ | エ | ア | オ | ウ | エ | | | | | |
| 問２ | エ | オ | エ | エ | | | | | | |
| 問３ | ウ | ウ | エ | エ | | | | | | |
| 問４ | オ | オ | カ | エ | エ | | | | | |
| 問５ | ウ | ア | ウ | オ | イ | | | | | |

＜解説＞

問１

【解説】

現在の使用されているほとんどのパソコンでは，シフト JIS コードが使用されており，事実上の標準となっている。JIS 漢字コードとは異なり，半角文字（1 バイト文字セット）と漢字が混在していても，1 バイト目で識別可能であり，日本国内で使用する分には支障はない。しかし，米国が発祥の地であるインターネット上では，ASCII 文字（7 ビットコード：0x00～0x7F）が標準であり，8 ビットすべてを使用するシフト JIS コードで書かれたメールなどでは，文字化けなどの問題が発生する可能性が高い（半角カタカナも同様）。

この点，JIS 漢字コードは，1 バイト目と 2 バイト目ともに，0x21～0x7E の範囲しか使用しておらず，それぞれの先頭 1 ビットを無視されても問題ない（常にビット'0'）。このため，インターネットの電子メールには，一般に JIS 漢字コードが使用されている（メールソフトが，自動的に JIS 漢字コードに変換して送信している）。

なお，JIS 漢字コードでは，ビットパターンから漢字であるかどうかを識別できないため，特別な符号 0x1B(エスケープシーケンス)を挿入して，漢字の始まりと終わりを示している。

設問１

a ：シフト JIS コードの 2 バイト目を判定する部分である。シフト JIS コードの漢字であれば，2 バイト目は 0x40～0x7E 及び 0x80～0xFC の範囲にある。この範囲にあるとき，真を返すようにすればよい。

ア　(c)>0x39 && (c)<=0x7E || (c)>0x79 && (c)<=0xFC

0x40-1は，0x39ではなく0x3Fであるため，判定する範囲が誤りになる。

イ　(c)<0x40 || (c)>0x7E && (c)<0x80 || (c)>0xFC

2バイト目が0x40～0x7E，及び0x80～0xFCの範囲にないとき真になる。

ウ　!((c)<0x40 && (c)>0x7E || (c)<0x80 && (c)>0xFC)

(c)<0x40 && (c)>0x7Eと(c)<0x80 && (c)>0xFCは常に偽となり，その否定を

取るため，結果は常に真になる。

エ　`(c)>=0x40 && (c)<=0xFC && (c)!=0x7F`

`0x7E`と`0x80`の間には，`0x7F`のみが存在する。したがって，`0x40`～`0xFC` の範囲内において，`0x7F`のみを除外すればよいので，これが正解となる。

オ　`(c)>=0x40 && (c)<=0xFC || (c)!=0x7F`

すべての範囲を示すことになるため誤りとなる。

b：問題文の“シフト JIS コードの全角文字を，JIS 漢字コードに変換するアルゴリズム”の②に対応する処理である。「1 バイト目から `0x81` を引き，1 ビット左へシフトする」ので，`(*b1 - 0x81) << 1` でよい。このとき，減算の「-」とビットシフトの「<<」では，演算の優先順位は「-」の方が高い。したがって，`(*b1 - 0x81)`の`( )`は不要であることから，「`*b1 - 0x81 << 1`」が正解である。

c：`while` ループに入る前に行うべき処理は，変数 `flg` の初期値設定である。変数 `flg` は，`if` 文で参照されており，真(TRUE)の場合は，漢字 IN のエスケープシーケンスを，偽(FALSE)の場合は，漢字 OUT のエスケープシーケンスを出力している。最初にシフト JIS コードの漢字が現れたとき，漢字 IN のエスケープシーケンスを出力する必要があるため，ここでは，初期値として真（0 以外）を設定しておくべきである。

なお，漢字が連続する場合は，ASCII コードの文字が現れるまで，漢字 OUT のエスケープシーケンスを出力する必要はないため，`if(flg)`の内部で `flg` に 0 を代入している。

d：関数 `sjis_jis` では，引数として`(int *b1, int *b2)`が指定されている。これは，`int` 型へのポインタを表しており，関数内でポインタが指す領域の値を直接更新している。このため，呼び出し側では，値が格納されている int 型変数のアドレスを指定しなければならない。指定方法は，「`sjis_jis(&c, &d)`」である。

**設問２**

シフト JIS コードの「あ」（`0x82A0`）を，問題文で示されているアルゴリズムにしたがって，JIS 漢字コードに変換する。（1 バイト目＝`0x82`, 2 バイト目＝`0xA0`）

① 1 バイト目が `0xE0` 以上であれば，1 バイト目から `0x40` を引く。
　→ 何もしない

② 1 バイト目から `0x81` を引き，1 ビット左へシフトする。
　→ `0x82−0x81<<1` を行うと，1 バイト目は `0x02` となる。

③ 2 バイト目が `0x80` 以上であれば，2 バイト目から 1 を引く。
　→ `0xA0−1` を行うと，2 バイト目は `0x9F` となる。

④ 2 バイト目が `0x9E` 以上であれば，1 バイト目に 1 を加算し，2 バイト目から `0x9E` を引く。そうでなければ，2 バイト目から `0x40` を引く。
　→ 2 バイト目は `0x9E` 以上であるから，1 バイト目 `0x02＋1→0x03`，2 バイト目

0x9F－0x9E→0x01 となる。

⑤　1バイト目と2バイト目に，それぞれ 0x21 を加算する。

→　1バイト目 0x03＋0x21→0x24，2バイト目 0x01＋0x21→0x22 となる。したがって，JIS 漢字コードの「あ」は，0x2422 であることがわかる。

〔 メ モ 用 紙 〕

〔　メ　モ　用　紙　〕

〔 メ モ 用 紙 〕