/* ___ *

src2tex* version 2.12 (Sep 9, 1996)

KAZUO AMANO† AND SHINICHI NOMOTO‡

* ___ */

/* ___ pas_bold.c ___ */
/* In PASCAL, several key words are usually printed in bold face. Our **src2tex** and **src2latex** print those key words in bold typewriter fonts if the input file is written in PASCAL. */

```
#include <stdio.h>
#include "src2tex.h"

extern int *dec_buf_ptr();
extern int *inc_buf_ptr();
```

/* ___ table of key words of PASCAL ___ */
/* If input source file is written in PASCAL, then the following words are printed in bold face. */

```
char *PAS_Table[] =
{"absolute", "and", "array",
 "begin",
 "case", "const",
 "define", "div", "downto", "do",
 "else", "end", "external",
 "file", "forward", "for", "function",
 "goto",
 "if", "inline", "in",
 "label", "let",
 "mod",
 "nil", "not",
 "of", "or", "out",
 "packed", "procedure", "program",
 "record", "repeat", "return",
 "set", "shl", "shr", "string",
 "then", "to", "type",
 "until",
 "var",
 "while", "with",
 "xor",
```


† Department of Mathematics, Josai University, Keyakidai, Sakado, Saitama, 350-02, JAPAN (kamano@po.iijnet.or.jp)

‡ Department of Mathematics, Josai University, Keyakidai, Sakado, Saitama, 350-02, JAPAN (m93093@crescent.josai.ac.jp)

```
 ""};
```

/* ——— compare two words ——— */
/* This function compares two strings

buf_ptr=*string1*, tbl_ptr=*string2*

and returns

$$\texttt{flag}\begin{cases} =0 & \text{if } string1{=}string2 \\ \neq 0 & \text{otherwise .}\end{cases}$$

However, compare_word() is slightly different from strcmp(). */

```
int compare_word(buf_ptr, tbl_ptr)
int *buf_ptr;
char *tbl_ptr;
{
    char c;
    int *ptr, flag = 0;

    /* if the previous character is either an alphabet or a number, then returns -1 */
    ptr = dec_buf_ptr(buf_ptr);
    c = *ptr;
    if ((c >= '0') && (c <= '9'))
        return (-1);
    if ((c >= 'A') && (c <= 'Z'))
        return (-1);
    if ((c >= 'a') && (c <= 'z'))
        return (-1);
    if (c == '_')
      return (-1);
    /* flag ≠ 0 if two strings are different */
    /* flag = 0 otherwise */
    ptr = buf_ptr;
    while (*tbl_ptr != '\0')
      {
          c = *ptr;
          /* upper case -> lower case */
          if ((c >= 'A') && (c <= 'Z'))
              c += 0x20;
          flag = c - *tbl_ptr;
          if (flag != 0)
              break;
          ptr = inc_buf_ptr(ptr);
          ++tbl_ptr;
      }
    /* if the next character is either an alphabet or a number, then returns -1 */
    c = *ptr;
    if ((c >= '0') && (c <= '9'))
        return (-1);
    if ((c >= 'A') && (c <= 'Z'))
        return (-1);
    if ((c >= 'a') && (c <= 'z'))
        return (-1);
    if (c == '_')
```

```
      return (-1);
    return flag;
}
```

/* ——————————— get bold face flag ——————————— */
/* This function compares a string

buf_ptr=*string*

with key words stored in `PAS_Table[]`. If there exists a key word which is equal to given *string*, then it returns the length of *string*. */

```
int get_bf_flag(buf_ptr)
int *buf_ptr;
{
    char **table, *ptr;
    int i = 0;

    for (table = PAS_Table; **table != '\0'; ++table)
      {
        if (compare_word(buf_ptr, *table) == 0)
          {
#ifdef DEBUGGING
            printf("compare_word(): ");
            printf("a reserved word \"%s\" is found in PAS_table\n", *table);
#endif
            /* strlen(*table)                                        */
            for (i= 0, ptr = *table; *ptr != '\0'; ++i, ++ptr);
            break;
          }
      }
    return i;
}
```