```
/*
 * include/crt0.hmm : lib/crt0.s の外部インタフェース
 */
public int    _aToI(void[] a);                      // 型変換ルーチン
public void[] _iToA(int a);                          // 型変換ルーチン
public void[] _aToA(void[] a);                       // 型変換ルーチン
public void[] _addrAdd(void[] a, int i);             // アドレス計算
public int    _uCmp(int a, int b);                   // 符号無しの大小比較
public int    _aCmp(void[] a, void[] b);             // アドレスの大小比較
public void[] _args();                               // 関数の引数配列を返す
public int[]  _add32(int[] dst, int[] src);          // 32ビット加算
public int[]  _sub32(int[] dst, int[] src);          // 32ビット減算
public int[]  _mul32(int[] dst, int src);            // 32ビット乗算
public int[]  _div32(int[] dst, int src);            // 32ビット除算
public int    _mod32(int[] dst, int src);            // 32ビット剰余算
public int    _sp();                                 // 現在の SP の値を返す
public int    _in(int p);                            // I/Oからワードを入力
public void   _out(int p, int v);                    // I/Oへワードを出力


/*
 * include/ctype.hmm : lib/ctype.cmm の外部インタフェース
 */
public boolean isAlnum(char c);
public boolean isAlpha(char c);
public boolean isDigit(char c);
public boolean isLower(char c);
public boolean isPrint(char c);
public boolean isSpace(char c);
public boolean isUpper(char c);
public boolean isXdigit(char c);
public char toLower(char c);
public char toUpper(char c);


/*
 * include/stdio.hmm : lib/stdio.cmm, lib/printf.cmm の外部インタフェース
 */
public boolean feof(FILE stream);
public boolean ferror(FILE stream);
public boolean fflush(FILE stream);
public boolean fseek(FILE stream, int h, int l);
public boolean fsize(char[] path, int[] size);
public FILE fopen(char[] name, char[] mode);
public boolean fclose(FILE stream);
public boolean fputc(char c, FILE stream);
public boolean putchar(char c);
public boolean fputs(char[] s, FILE stream);
public boolean puts(char[] s);
public char fgetc(FILE stream);
public char getchar();
public char[] fgets(char[] s, int n, FILE stream);
public void perror(char[] s);
public int fprintf(FILE stream, char[] format, ...);
public int printf(char[] format, ...);
```

```
/*
 * include/stdlib.hmm : lib/stdlib.cmm の外部インタフェース
 */
public void[] malloc(int s);
public void[] calloc(int c, int s);
public void free(void[] m);
public int atoi(char[] s);
public int htoi(char[] s);
public int rand();
public void srand(int seed);
public void exit(int status);




/*
 * include/string.hmm : lib/string.cmm の外部インタフェース
 */
public char[] strCpy(char[] d, char[] s);
public char[] strNcpy(char[] d, char[] s, int n);
public char[] strCat(char[] d, char[] s);
public char[] strNcat(char[] d, char[] s, int n);
public int strCmp(char[] s1, char[] s2);
public int strNcmp(char[] s1, char[] s2, int n);
public int strLen(char[] s);
public int strChr(char[] s, char c);
public int strRchr(char[] s, char c);
public int strStr(char[] s1, char[] s2);




/*
 * wrapper.h : wrapper.c 関数のプロトタイプ宣言
 */
inline static void *_addrAdd(void *a, int inc) { ... }
inline static void *_aToA(void *a) { ... }
inline static long lToL(unsigned int l[]) { ... }
void * _mAlloc(int s);
FILE *_fOpen(char *n, char *m);
int _feof(FILE *fp);
int _fseek(FILE *stream, int offsh, int offsl);
int fsize(char *path, int *size);
int htoi(char *s);
int _fPrintf(FILE *fp, _CA* fmt, ...);
int _printf(_CA* fmt, ...);

#ifdef _RTC
inline static FILE* __fOpen(_CA* path, _CA* mode) { ... }
inline static _CA* _fgets(_CA* buf, int n, FILE* fp) { ... }
inline static int _fputs(_CA* buf, FILE *fp) { ... }
inline static int _puts(_CA* buf) { ... }
inline static void _perror(_CA* buf) { ... }
inline static int _fsize(_CA* path, _IA* size) { ... }
inline static int _atoi(_CA* str) { ... }
inline static int _htoi(_CA* str) { ... }
#endif
```

```
/*
 * lib/ctype.cmm : 文字分類など
 */
public boolean isSpace(char c) { ... }
public boolean isDigit(char c) { ... }
public boolean isXdigit(char c) { ... }
public boolean isPrint(char c) { ... }
public boolean isLower(char c) { ... }
public boolean isUpper(char c) { ... }
public boolean isAlpha(char c) { ... }
public boolean isAlnum(char c) { ... }
public char toLower(char c) { ... }
public char toUpper(char c) { ... }




/*
 * lib/string.cmm : 文字列操作
 */
public char[] strCpy(char[] d, char[] s) { ... }
public char[] strNcpy(char[] d, char[] s, int n) { ... }
public char[] strCat(char[] d, char[] s) { ... }
public char[] strNcat(char[] d, char[] s, int n) { ... }
public int strCmp(char[] s1, char[] s2) { ... }
public int strNcmp(char[] s1, char[] s2, int n) { ... }
public int strLen(char[] s) { ... }
public int strChr(char[] s, char c) { ... }
public int strRchr(char[] s, char c) { ... }
public int strStr(char[] s1, char[] s2) { ... }




/*
 * printf.cmm : TacOS 用の入出力ルーチン
 */
public int fprintf(FILE fp, char[] fmt, ...) { ... }
public int printf(char[] fmt, ...) { ... }




/*
 * lib/stdio.cmm : TacOS 用の入出力ルーチン
 */
public void _stdioInit() { ... }
public boolean fflush(FILE fp) { ... }
public void _flushAll() { ... }
public boolean fputc(char c, FILE fp) { ... }
public boolean putchar(char c) { ... }
public boolean fputs(char[] str, FILE fp) { ... }
public boolean puts(char[] str) { ... }
public boolean feof(FILE fp) { ... }
public char fgetc(FILE fp) { ... }
public char getchar() { ... }
public char[] fgets(char[] str, int n, FILE fp) { ... }
public boolean fseek(FILE fp, int offsh, int offsl){ ... }
public boolean fsize(char[] path, int[] size) { ... }
public FILE fopen(char[] name, char[] mode) { ... }
public boolean fclose(FILE fp) { ... }
public boolean ferror(FILE fp) { ... }
public void _perror(char[] msg, int err) { ... }
public void perror(char[] msg) { ... }
public int readDir(int fd, Dir dir) { ... }
```

```
/*
 * lib/stdlib.cmm : C-- のライブラリ関数
 */
public void _stdlibInit() { ... }
public void[] malloc(int siz) { ... }
public void free(void[] mem) { ... }
public void[] calloc(int n, int s) { ... }
public int atoi(char[] s) { ... }
public int htoi(char[] s) { ... }
public void exit(int status) { ... }
public void srand(int seed) { ... }
public int rand() { ... }



/*
 * printf.c : C言語版のためのラッパー
 */
#ifdef _RTC
int _fPrintf(FILE *fp, _CA* fmt, ...) { ... }
int _printf(_CA* fmt, ...) { ... }
#else
int _fPrintf(FILE *fp, char* fmt, ...) { ... }
int _printf(char* fmt, ...) { ... }
#endif



/*
 * wrapper.c : C-- 版と C 版で仕様が異なる関数など
 *             (主に stdlib.hmm の関数)
 */
#ifdef _RTC
int __main(int argc, _RA* args, _RA* envs);
int main(int argc, char *argv[], char *envp[]) { ... }
#endif

void * _mAlloc(int s) { ... }
FILE *_fOpen(char *n, char *m) { ... }
int _feof(FILE *fp) { ... }
int _fseek(FILE *stream, int offsh, int offsl){ ... }
int fsize(char *path, int *size) { ... }
int htoi(char *s) { ... }



/*
 * lib/cfunc.hmm : C-- 関数と C 関数の読みかえを定義
 */
#define malloc(s) _mAlloc(s)
#define feof(fp) _feof(fp)
#define fseek(s,h,l) _fseek(s,h,l)
#define fprintf    _fPrintf
#define printf     _printf

#ifdef _RTC
#define main __main
#define fopen(p,m)   __fOpen(p,m)
#define fgets(b,n,f) _fgets(b,n,f)
#define fputs(s,f)   _fputs(s,f)
#define puts(s)      _puts(s)
#define perror(s)    _perror(s)
#define fsize(p,s)   _fsize(p,s)
#define atoi(s)    _atoi(s)
#define htoi(s)    _htoi(s)
#else
#define fopen(p,m) _fOpen(p,m)
#endif
```