EIN- AUSGABE <stdio.h>

Dateioperationen

```
FILE *fopen(const char *filename,const char *mode)
FILE *freopen(const char *filename,const char *mode,FILE *stream)
int fflush(FILE *stream)
int fclose(FILE *stream)
int remove(const char *filename)
int rename(const char *oldname,const char *newname)
FILE *tmpfile(void)
char *tmpnam(char s[L_tmpnam])
int setvbuf(FILE *stream,char *buf,int mode,size_t size)
void setbuf(FILE *stream,char *buf)
```

Formatierte Ausgabe

```
int fprintf(FILE *stream,const char *format,...)
int printf(const char *format,...)
int sprintf(char *s,const char *format,...)
vprintf(const char *format,va_list arg)
vfprintf(FILE *stream,const char *format,va_list arg)
vsprintf(char *s,const char *format,va_list arg)
```

Formatierte Eingabe

```
int fscanf(FILE *stream,const char *format,...)
int scanf(const char *format,...)
int sscanf(char *s,const char *format,...)
```

Zeichenweise Ein- Ausgabe

```
int fgetc(FILE *stream)
char *fgets(char *s,int n,FILE *stream)
int fputc(int c,FILE *stream)
int fputs(const char *s,FILE *stream)
int getc(FILE *stream)
int getchar(void)
char *gets(char *s)
int putc(int c, FILE *stream)
int putchar(int c)
int puts(const char *s)
int ungetc(int c,FILE *stream)
```

Direkte Ein- Ausgabe

```
size_t fread(void *ptr,size_t size,size_t nobj,FILE *stream)
size_t fwrite(const void *ptr,size_t size,size_t nobj,FILE *stream)
```

Dateipositionierung

```
int fseek(FILE *stream,long offset,int origin)
long ftell(FILE *stream)
void rewind(FILE *stream)
int fgetpos(FILE *stream,fpos_t *ptr)
int fsetpos(FILE *stream,const fpos_t *ptr)
```

Fehler/Dateiende

```
void clearerr(FILE *stream)
int feof(FILE *stream)
int ferror(FILE *stream)
void perror(const char *s)
```

KLASSIFIZIERUNG VON ZEICHEN <ctype.h>

```
int isalnum(int c)
int isalpha(int c)
int iscntrl(int c)
int isdigit(int c)
int isgraph(int c)
int islower(int c)
int isprint(int c)
int ispunct(int c)
int isspace(int c)
int isupper(int c)
int isxdigit(int c)
int tolower(int c)
int toupper(int c)
```

MANIPULATION VON ZEICHENKETTEN(STRINGS) <string.h>

```
char *strcpy(char *s,const char *t)
char *strncpy(char *s,const char *t,size_t n)
char *strcat(char *s,const char *t)
chat *strncat(char *s,const char *t,size_t n)
int strcmp(const char *s,const char *t)
int strncmp(const char *s,const char *t,size_t n)
char *strchr(const char *s,int c)
char *strrchr(const char *s,int c)
size_t strspn(const char *s,const char *t)
size_t strcspn(const char *s,const char *t)
char *strpbrk(const char *s,const char *t)
char *strstr(const char *s,const char *t)
size_t strlen(const char *s)
char *strerror(size_t n)
char *strtok(char *s,const char *t)
void *memcpy(void *s,const void *t,size_t n)
void *memmove(void *s,const void *t,size_t n)
int memcmp(const void *s,const void *t,size_t n)
void *memchr(const void *s,int c,size_t n)
void *memset(void *s,char c,size_t n)
```

MATHEMATISCHE FUNKTIONEN <math.h> -lm

```
double sin(double x)
double cos(double x)
double tan(double x)
double asin(double x)
double acos(double x)
double atan(double x)
double atan2(double x,double y)
double sinh(double x)
double cosh(double x)
double tanh(double x)
double exp(double x)
double log(double x)
double log10(double x)
double pow(double x,double y)
double sqrt(double x)
double ceil(double x)
double floor(double x)
double fabs(double x)
double ldexp(double x,int n)
double frexp(double x,int *exp)
double modf(double x,double *ip)
double fmod((double x,double y)
```

DIENSTFUNKTIONEN <stdlib.h>

```
double atof(const char *s)
int atoi(const char *s)
long atol(const char *s)
double strtod(const char *s,char **endp)
long strtol(const char *s,char **endp,int base)
unsigned long strtoul(const char *s,char **endp,int base)
int rand(void)
void srand(unsigned int seed)
void *calloc(size_t nobj,size_t size)
void *malloc(size_t size);
void *realloc(void *p,size_t size)
void free(void *p)
void abort(void)
void exit(int status)
int atexit(void (*fcn)(void))
int system(const char *s)
char *getenv(const char *name)
void *bsearch(const void *key, const void *base, size_t n,size_t size,
              int (*cmp)(const void *kv,const void *dt))
void *qsort(void *base,size_t n, size_t size,
              int (*cmp)(const void *kv,const void *dt))
int abs(int n)
long labs(long n)
div_t div(int num,int denom)
ldiv_t ldiv(long num,long denom)
```

DIAGNOSTICS <assert.h>

```
void assert(int expr)
```

VARIABLE ARGUMENTLISTEN <stdarg.h>

```
va_start(va_list ap,lastarg)
type va_arg(va_list ap,type)
void va_end(va_list ap)
```

NICHTLOKALE SPRUENGE <setjmp.h>

```
int setjmp(jmp_buf env)
void longjmp(jmp_buf env,int val)
```

SIGNALE <signal.h>

```
void (*signal(int sig,void (*handler)(int)))(int)
int raise(int sig)
```

ZEIT- UND DATUMSFUNKTIONEN <time.h>

```
clock_t clock(void)
time_t time(time_t *tp)
double difftime(time_t time2,time_t time1)
time_t mktime(struct tm *tp)
char *asctime(const struct tm *tp)
char *ctime(const time_t *tp)
struct tm *gmtime(const time_t *tp)
struct tm *localtime(const time_t *tp)
size_t strftime(char *s,size_t smax,const char *fmt,const struct tm *tp)
```

IMPLEMENTATIONSABHÄNGIGE WERTE <limits.h>