

COMANDOS BÁSICOS

Diferente do Pascal, o C tem funções e constantes dentro de algumas bibliotecas. Portanto, vejamos as principais bibliotecas, suas funções e constantes.

STDIO.H

FUNÇÃO	DESCRIÇÃO
FILE	Variável do tipo arquivo.
fclose(FILE *stream)	Fecha uma stream (arquivo).
feof(FILE *stream)	Testa o fim de um arquivo.
fgetc(FILE *stream)	Lê um caracter de uma stream.
fputc(int c, FILE *stream)	Escreve um caracter em uma stream.
fgetpos(FILE *stream, fpos_t *pos)	Retorna a posição do ponteiro na stream.
fsetpos(FILE *stream, const fpos_t *pos)	Ajusta o ponteiro de uma stream.
fgets(char *s, int n, FILE *stream)	Lê n caracteres de uma stream para a string s.
fputs(const char *s, FILE *stream)	Copia a string s para a stream.
fopen(const char *filename, const char *mode)	Abre uma stream. Modos, adiante.
printf()	Funções de escrita. Vide tabela abaixo.
scanf()	Funções de leitura. Idem tabela printf.
fread(void *ptr, size_t size, size_t n, FILE stream)	Transfere um bloco da stream para um buffer.
fwrite(void *ptr, size_t size, size_t n, FILE stream)	Transfere um bloco do buffer para uma stream.
fseek(FILE *stream, long offset, int whence)	Ajusta ponteiro de offset posições, a partir de whence (vide abaixo).
EOF	Indica fim de arquivo.

Modos de abertura de arquivo:

Modo	Descrição
r	Somente leitura.
w	Criado para escrita.
a	Append (continua arquivo existente).
r+	Abre um arquivo existente para atualização (escrita e leitura).
w+	Cria um novo arquivo para atualização.
a+	Append, se existente, ou cria, se não existente.

t	Arquivo texto.
b	Arquivo binário

Exemplo:

```
fopen("teste.bin","rb");
```

Whence do fseek

Constante	Descrição
SEEK_SET	A partir do início do arquivo. Valor de SEEK_SET é 0.
SEEK_CUR	A partir da posição atual. Valor de SEEK_CUR é 1.
SEEK_END	A partir do fim do arquivo. Valor de SEEK_END é 2.

Funções relacionadas ao printf:

Função	Descrição
cprintf	Envia saída formatada para a tela.
fprintf	Envia saída formatada para uma stream.
printf	Envia saída formatada para a tela.
sprintf	Envia saída formatada para uma string.
vfprintf	Envia saída formatada para uma stream, usando uma lista de argumentos.
vprintf	Envia saída formatada para a tela, usando uma lista de argumentos.
vsprintf	Envia saída formatada para uma string, usando uma lista de argumentos.

Exemplo de abertura de arquivo binário:

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

```
FILE *arquivo;  
char buffer[1024];
```

```

void main(void)
{
    arquivo = fopen("exemplo.bin","rb");
    fread(buffer, 1024, 1, arquivo);
    fseek(arquivo,40,SEEK_SET);
    fclose(arquivo);
}

```

Para saber se tal arquivo que está sendo aberto não foi encontrado, faça o seguinte teste:

```

if ( (arquivo = fopen("exemplo.bin","rb")) == NULL)
    printf("Não achei teste arquivo.");

```

MATH.H

Função	Descrição
<code>fabs(double)</code>	Absoluto. <code>abs</code> (inteiro), <code>cabs</code> (complexo), <code>fabs</code> (real), <code>labs</code> (longo).
<code>floor(double)</code>	Trunca número real.
<code>ceil(double)</code>	Arredonda número real
<code>modf(double, double *)</code>	Quebra um real em uma parte inteira e outra fracionária.
<code>sqrt(double)</code>	Raiz quadrada.
<code>atof(char *)</code>	Converte string para real.
<code>sin(double)</code>	Seno.
<code>cos(double)</code>	Cosseno.
<code>tan(double)</code>	Tangente.
<code>asin(double)</code>	Arco seno.
<code>acos(double)</code>	Arco cosseno.
<code>atan(double)</code>	Arco tangente.
<code>atan2(double, double)</code>	Arco tangente.
<code>frexp(double, int *)</code>	Quebra número real em mantissa e expoente.
<code>ldexp(double, int)</code>	Equivale a $(double * 2^{int})$
<code>log(double)</code>	Logaritmo natural.
<code>log10(double)</code>	Logaritmo na base 10.
<code>pow(double, double)</code>	$x ^ y$. Ex: <code>pow(2,4) = 16</code> .
<code>exp(double)</code>	$e ^ double$
<code>sinh(double)</code>	Seno hiperbólico.

cosh(double)
tanh(double)

Cosseno hiperbólico.
Tangente hiperbólico.

Exemplo de utilização das funções matemáticas:

```
#include <stdio.h>
#include <math.h>

void main(void)
{
    double angulo= 1.45;
    printf("O seno de %f radianos é: %f",angulo,sin(angulo));
}
```

STDLIB.H

Função	Descrição
atof(char *)	Converte string para real.
atoi(char *)	Converte string para inteiro.
atol(char *)	Converte string para longo.
rand(void)	Gera número aleatório. Ex: rand() % 100.
srand(unsigned int)	Inicializa gerador de números aleatórios.
calloc(size_t, size_t)	Aloca memória principal.
free(void *)	Libera memória alocada.
malloc(size_t)	Aloca memória.
realloc(void *, size_t)	Realoca memória principal.
abort(void)	Termina um processo.
exit(int)	Termina um programa.
getenv(char *), putenv(char *)	Pega uma string do meio. Coloca no meio uma string.
system(char *)	Realiza um comando de DOS. Ex: system("dir").
qsort(void *, size_t, size_t, int (*)(void *, void *))	Realiza uma ordenação por quicksort.
abs(int)	Valor absoluto, isto é, sem o sinal; módulo. Para inteiros
labs(long)	Valor absoluto, isto é, sem o sinal; módulo. Para longos.

Exemplo de número aleatório:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void)
{
    int i;
    time_t time;
    srand((unsigned) time(&t));
    printf("Dez números aleatórios entre 0 e 99\n\n");
    for (i=0; i<10; i++)
        printf("%d\n", rand() % 100);
    return 0;
}
```

Exemplo de alocação de memória:

```
#include <stdlib.h>

char *buffer;

void main(void)
{
    buffer = (char *) malloc(10);
}
```

Buffer fica com tamanho de 10 bytes. A cada malloc, n bytes são acrescentados ao tamanho de buffer.

Exemplo de quicksort:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
int ordenacao( const void *a, const void *b);
```

```
char lista[5][10] = {"Gato", "Cachorro", "Papagaio", "Rato",
"Ramster"};
```

```
void main(void)
{
    int x;
    qsort((void *)lista, 5, sizeof(lista[0]),ordenacao);
    for(x=0; x<5; x++)
        printf("%s\n",lista[x]);
}
```

STRING.H

Função	Descrição
memcpy(void *, void *, size_t)	Copia um bloco de n bytes para destino de fonte. memcpy(dest, fonte, n).
memmove(void *, void *, size_t)	Copia um bloco de n bytes para destino de fonte. memmove(dest, fonte, n).
strcpy(char *, char *)	Copia string. Sempre destino, fonte.
strncpy(char *, char *, size_t)	Copia n caracteres.
strcat(char *, char *)	Concatena string.
strncat(char *, char *, size_t)	Concatena n caracteres.
memcmp(void *, void *, size_t)	Compara duas strings.Retorna 0, se foram iguais.
strcmp(char *, char *)	Compara duas strings.Retorna 0, se foram iguais.
strncmp(char *, char *, size_t)	Compara duas strings, em n caracteres.
strcoll(char *, size_t, char *)	Compara duas strings.
memchr(void *, int, size_t)	Procura n bytes pelo caracter c.
strcspn(char *, char *)	Procura em que parte duas string tem pontos em comum.
strpbrk(char *, char *)	Procura em uma string, a primeira ocorrência de qualquer caracter da segunda.

<code>strspn(char *, char *)</code>	Procura em que parte duas string passam a se diferenciar.
<code>strstr(char *, char *)</code>	Acha a primeira ocorrência de uma substring em uma string.
<code>strtok(char *, char *)</code>	Procura na primeira string o primeiro token não encontrado na segunda.
<code>memset(void *, int, size_t)</code>	Ajusta para menos, o tamanho de uma string.
<code>strerror(int)</code>	Constrói uma mensagem de erro.
<code>strlen(char *)</code>	Retorna o tamanho de uma string.
<code>strchr(char *, int)</code>	Procura a primeira ocorrência de um caracter.
<code>strrchr(char *, int)</code>	Procura a última ocorrência de um caracter.

Exemplo de concatenação:

```
#include <stdio.h>
#include <string.h>
```

```
char *pal1 = "arara", *pal2 = "quara";
```

```
void main(void)
{
    strcat(pal1,pal2);
    printf("Pal1 agora é %s",pal1);
}
```

ONDE ENCONTRAR DOCUMENTAÇÃO COMPLETA

O Turbo C++ 3 da Borland, ou o C++ Builder 3 (encontrado em revistas) tem um help, detalhando cada função e mostrando diversos exemplos de como se usar tais funções. Para procurar as funções relativas ao `math.h`, procure por `math.h`. Dentro dele, estão todas as funções a ela relacionadas. Escolha uma e veja sua descrição, com um exemplo.

Para certificar-se que o C do MSX tem ou não esta função, abra o respectivo arquivo `.h` do MSX, em um editor de texto. Por exemplo,

a função `fclose` é encontrada no `math.h` do MSX:

```
extern int fclose(FILE *);
```

Portanto, eu posso utilizar esta função no MSX.

[/MARMSX/CURSOS/C--](#)