

stdlib.h

Conversão de tipo	
atof	string para float
atoi	string para int
atol	string para long int
strtod	string para double
strtoul	string para long int
strtol	string para long int
stroul	string para unsigned long int
ltoa	Int para string
Geração de sequência pseudoaleatória	
rand	gera um número inteiro pseudoaleatório
srand	seleciona a semente do gerador pseudoaleatório
Alocação e liberação de memória	
malloc calloc realloc	aloca memória
free	libera memória
Controle de processos	
abort	força o término da execução
atexit	registra uma função "callback" para saída do programa
exit	termina a execução do programa
getenv	obtem uma variável de ambiente
system	executa um comando externo
Ordenamento e procura	
bsearch	procura binária em "array"
qsort	ordena "array" segundo algoritmo Quick Sort
Matemática	
abs labs	Valor absoluto
div ldiv	divisão inteira
Ordenação	
qsort	Função QuickSort.

qsort (void *base, size_t num, size_t size, int (*compar) (const void*, const void*))	
Base	array a ser ordenado
Num	Tamanho do array
Size	Tamanho, em bytes de cada posição do array
Compar	<p>Ponteiro para a função de comparação (dizer como vai coparar os elementos para colocar na ordem correta)</p> <p>Função que define como dois valores (p1 e p2" do array deverão ser comparados. Deve seguir o protótipo:</p> <p>Int compar (const void *p1, const void *p2);</p> <p>Retorno da função:</p> <p>Valor menor que 0: p1 vem antes de p2 Valor igual a 0: p1 é igual a p2 Valor maior que 0: p1 vem depois de p2</p>
Size_t	Apenas um inteiro sem sinal

Exemplo:

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

int comparaCrescente (const void *a, const void *b) {
if (*(int*)a == *(int*)b) // cast. Primeiro asterisco acessa o conteudo
return 0; //iguais
else
if (*(int*)a < *(int*)b)
return -1; // vem antes
else
return 1; // vem depois
}

int main() {
int vet[10] = {0, 1, 9, 2, 8, 3, 7, 4, 6, 5};

qsort(vet, sizeof(vet)/sizeof(int), sizeof(int), comparaCrescente);

for (int i=0; i<10; i++)
printf("%d ", vet[i]);
}
```

stdio.h

printf()	Função usada para imprimir dados na tela
scanf()	Função usada para capturar dados do usuário

fprintf()	Função usada para imprimir dados em arquivo
fscanf()	Função usada para ler dados de arquivos

locale.h

LC_ALL	Sets everything.
LC_COLLATE	Affects strcoll and strxfrm functions.
LC_CTYPE	Affects all character functions.
LC_MONETARY	Affects the monetary information provided by localeconv function.
LC_NUMERIC	Affects decimal-point formatting and the information provided by localeconv function.
LC_TIME	Affects the strftime function.

time.h

char *asctime(const struct tm* tmptr)	Converte tm para uma string no formato "Www Mmm dd hh:mm:ss yyyy", onde Www é o dia da semana, Mmm o mês em letras, dd o dia do mês, hh:mm:ss a hora, e yyyy o ano. A string é seguida por uma nova linha e um caracter nulo (NULL), formada por um total de 26 caracteres. A string gerada é alocada estaticamente e compartilhada pelas funções ctime e asctime. Cada vez que uma dessas funções é chamada o conteúdo da string é substituído.
clock_t clock(void)	Retorna o tempo decorrido desde que o programa iniciou a execução.
char* ctime(const time_t* timer)	Converte o valor de time_t para uma string com um formato igual ao de asctime. Essa string gerada é alocada estaticamente e compartilhada pelas funções ctime e asctime. Cada vez que uma dessas funções é chamada, ou usada, o conteúdo da string é sub-escrito. asctime_também usa internamente o arquivo temporário usado por gmtime e localtime como valor de retorno, então uma chamada dessa função vai alterar esse buffer (ou arquivo temporário).
double difftime(time_t timer2, time_t timer1)	Retorna em segundos a diferença entre <u>dois</u> espaços de tempo.
struct tm* gmtime(const time_t* timer)	Converte um valor time_t para uma estrutura tm como o relógio UTC. Essa estrutura é alocada estaticamente e usada também pelas funções gmtime, localtime e ctime. Essa estrutura tem seu valor alterado cada vez que uma dessas funções é chamada.
struct tm* gmtime_r(const time_t* timer, struct tm* result)	Converte um valor time_t para uma estrutura tm igual ao UTC. O tempo é armazenado na estrutura tm referenciada pelo <i>resultado</i> . Essa função é uma versão segura da função gmtime.
struct tm* localtime(const time_t* timer)	Converte um valor time_t para uma estrutura tm, igual ao relógio local. Essa estrutura é alocada estaticamente e usada também pelas funções gmtime, localtime e ctime. Essa estrutura tem seu valor alterado cada vez que uma dessas funções é chamada.
time_t mktime(struct tm* ptm)	Converte tm para valor de tempo time_t. Checa os membros da estrutura tm passada como parâmetro <i>ptm</i> ajustando os valores se esses não estão no renga, ou tamanho certo ou estão incompletos ou errados e então traduz a estrutura para um valor <u>time_t</u> então retornado. O valor original do <i>dia da semana</i> ,

	membro do <code>tm_mdia</code> não é checado enquanto <code>tm_mes</code> e <code>tm_ano</code> estão determinados. Caso haja erro, o valor -1 é retornado.
<code>time_t time(time_t* timer)</code>	Pega a data atual, em segundos de acordo com o relógio Unix, a partir do relógio do sistema. Guarda esse valor no timer. Se o timer é igual a <code>null</code> , o valor não é armazenado, mas esse valor é continuamente retornado pela função.
<code>size_t strftime(char* s, size_t n, const char* format, const struct tm* tptr)</code>	Formata <code>tm</code> numa string data/hora cujo resultado é colocado no array <code>s</code> .

math.h

É importante lembrar que as funções que envolvem ângulos devem receber valores em **radianos** e não em **graus**.

Funções pré-C99	
<code>acos</code>	arco-coseno
<code>asin</code>	arco-seno
<code>atan</code>	arco-tangente
<code>atan2</code>	arco-tangente (círculo cheio)
<code>ceil</code>	Arredonda para cima
<code>cos</code>	co-seno
<code>cosh</code>	co-seno hiperbólico
<code>exp</code>	exponencial
<code>fabs</code>	valor absoluto (módulo de)
<code>floor</code>	Arredonda para baixo
<code>fmod</code>	resto de uma divisão
<code>frexp</code>	fracção norm./parte exp
<code>ldexp</code>	inverso de <code>frexp</code>
<code>log</code>	logaritmo natural
<code>log10</code>	logaritmo base 10
<code>modf(x,p)</code>	retorna a parte fraccionária de <code>x</code> e guarda a parte inteira para onde o apontador <code>p</code> aponta
<code>pow(x,y)</code>	retorna o resultado de <code>x</code> elevado a <code>y</code>
<code>sin</code>	seno
<code>sinh</code>	seno hiperbólico
<code>sqrt</code>	raiz quadrada
<code>tan</code>	tangente
<code>tanh</code>	tangente hiperólica

Funções C99	
acosh	arco-cosseno hiperbólico
asinh	arco-seno hiperbólico
atanh	arco-tangente hiperbólica
cbrt	raiz cúbica
copysign(x,y)	retorna o valor de x com o sinal de y
erf	função erro
erfc	função erro complementar
exp2(x)	calcula 2 elevado à x, 2^x
expm1(x)	subtrai da exponenciação x, $e^x - 1$
fdim(x,y)	diferença positiva entre x e y, $\text{fmax}(x-y, 0)$
fma(x,y,z)	multiplica e soma, $(x * y) + z$
fmax(x,y)	maior valor de x e y
fmin(x,y)	menor valor de x e y
hypot(x,y)	hipotenusa, $\text{sqrt}(x^2 + y^2)$
ilogb	o expoente de uma dízima, convertido num int
lgamma	logaritmo natural do valor absoluto da função gamma
llrint	arredonda para inteiro (retorna long long) usando o modo de arredondamento corrente
lrint	arredonda para inteiro (retorna long) usando o modo de arredondamento corrente
llround	arredonda para inteiro (retorna long long)
lround	arredonda para inteiro (retorna long)
log1p(x)	logaritmo natural de $1 + x$
log2	logaritmo binário
logb	extrai o expoente da dízima infinita não periódica
nan(s)	retorna NaN , possivelmente usando um argumento "string"
nearbyint	arredonda uma dízima para o inteiro mais próximo
nextafter(x,y)	retorna o próximo valor representável depois de x (towards y)
nexttoward(x,y)	o mesmo que nextafter, excepto que y é sempre um long double
remainder(x,y)	calcula o resto como exigido pelo IEC 60559
remquo(x,y,p)	o mesmo que remainder, mas guarda o quociente (como int) no alvo do apontador p
rint	arredonda para inteiro (retorna double) usando o modo de arredondamento corrente
round	arredonda para inteiro (retorna double), arredonda casa mais distante de zero

scalbln(x,n)	$x * FLT_RADIX^n$ (n é long)
scalbn(x,n)	$x * FLT_RADIX^n$ (n é int)
tgamma	função gamma
trunc	trunca uma dízima

ctype.h

tolower	Converte o caractere em minúsculo
toupper	Converte caractere minúsculo em maiúsculo.
isalnum	Verifica se o caractere é alfanumérico
isalpha	Verifica se o caractere é uma letra do alfabeto
iscntrl	Verifica se o caractere é um caractere de controle
isdigit	Verifica se o caractere é um dígito decimal
isgraph	Verifica se o caractere tem representação gráfica
islower	Verifica se o caractere é minúsculo
isprint	Verifica se o caractere é “imprimível”.
ispunct	Verifica se o caractere é um ponto
isspace	Verifica se o caractere é um espaço em branco
isupper	Verifica se o caractere é uma letra maiúscula
isxdigit	Verifica se o caractere é um dígito hexadecimal

conio.h

kbhit	kbhit();	Espera o usuário pressionar uma tecla qualquer
cgets	cgets(char);	Lê uma string
cscanf	cscanf(“%tipo”, &var);	Lê valores formatados, igual ao scanf
cprintf	printf(“%tipo”, var);	Formata valores e os escreve, igual ao printf
getche	getche();	Lê um char (Não precisa apertar enter)
getch	getch();	Lê um char (não precisa de enter e não mostra o caractere no console)

Interação com arquivos

fopen	Abre um arquivo
fclose	Fecha o arquivo garantindo a transferência do buffer.
fflush	Descarrega o buffer.
fscanf	Lê de um arquivo
fprintf	Escrita formatada através do arquivo
fgets	Obtém uma string do arquivo. <code>fgetc()</code> Obtém um caracter do arquivo.
fputs	Inserir uma string no arquivo.
fputc	Inserir um caracter no arquivo.
fread	Lê um bloco de dados do arquivo.
fwrite	Escreve um bloco de dados no arquivo.
fseek	para o início do arquivo.
rewind	Reposiciona o ponteiro. para o início do arquivo.
ftell	Retorna a posição do ponteiro

r	Abre o arquivo somente para leitura, a partir do início. O arquivo deve existir.
w	Cria um arquivo vazio para escrita. Se já havia o arquivo, ele é perdido. "a" Adiciona no final do arquivo. Se o arquivo não existir, a função o cria.
r+	Abre o arquivo para leitura e escrita, a partir do início. O arquivo deve existir.
w+	Cria um arquivo vazio para leitura e escrita. Se já havia o arquivo, ele é perdido
a+	Abre para adição ou leitura no final do arquivo. Se o arquivo não existir, a função o cria.

string.h

strcat	<code>strcat(string_destino, string_origem);</code>	Concatena duas strings
strncat	<code>strncat(string_destino, string_origem, tamanho);</code>	Concatena duas strings. O tamanho deve ser especificado
strcpy	<code>strcpy(string_destino, string_origem);</code>	Copia o conteúdo de uma string para outra.
strncpy	<code>strncpy(string_destino, string_origem, tamanho);</code>	Copia o conteúdo de uma string para outra. O tamanho deve ser especificado
strlen	<code>strlen(string);</code>	Retorna o tamanho de uma string.
strcmp	<code>strcmp(string1, string2);</code>	Compara o conteúdo de duas strings, diferenciando maiúsculas de minúsculas;

strcmpi	strcmpi(string1, string2);	Compara o conteúdo de duas strings não diferencia maiúsculas de minúsculas;
strchr	strchr(string, char);	Retorna um ponteiro na primeira ocorrência do char na string1
strrchr	strrchr(string, char);	Retorna um ponteiro da última ocorrência do caractere
strstr	strstr(string1, string2);	Retorna um ponteiro da primeira ocorrência da str2 na str1
strrstr	strrstr(string1, string2);	Retorna um ponteiro da última ocorrência da str2 na str1
strdup	char = strdup(string1);	Duplica a string
strlwr	strlwr(string);	Deixa a string minúscula
strupr	strupr(string);	Deixa a string maiúscula
strrev	strrev(string);	Inverte a string
strset	strset(string, char);	Muda todos os caracteres da string para o caractere escolhido
strnset	strnset(string, char, num)	Muda todos os caracteres da string para o caractere escolhido até certo ponto.
strtok	strtok("string", "delimitador")	Divide uma string utilizando um delimitador. Para pegar todos, utilize um while(token != NULL), onde token é um ponteiro.

Extras

%s	Exibe uma string
%c	Exibe um caractere
%d	Exibe um inteiro em formato decimal
%i	Exibe um inteiro
%e	Exibe um número em notação científica (com e minúsculo)
%E	Exibe um número com notação científica (com E minúsculo)
%f	Exibe um ponto flutuante em formato decimal
%o	Exibe um número em notação octal
%s	Exibe uma string
%u	Exibe um decimal sem sinal
%x	Exibe um número em hexadecimal com letras minúsculas
%X	Exibe um número em hexadecimal com letras maiúsculas
%%	Exibe um sinal de %

%p	Exibe um ponteiro
%lf	Exibe um double
\b	retrocesso
\f	Alimentação de formulário
\n	Quebra de linha
\r	Cursor vai pro inicio da linha
\t	Paragrafo (TAB)
\"	Aspas
\'	Apóstrofo
\0	Nulo
\\	Barra invertida
\a ou \7	Sinal sonoro

puts()	Imprime uma <u>string</u> na tela (não suporta formatação)
putchar()	Imprime um <u>char</u> na tela (não suporta formatação)

Comandos do sistema

system ("time/t") ;	Mostra a hora do pc
system ("date/t");	Mostra a data do pc
system ("color XX");	Muda a cor do console, onde XX é o código da cor.
system ("ver") ;	Mostra o sistema operacional em que está rodando
system ("vol") ;	Mostra informações do volume
system ("mem") ;	Verifica a memora
system ("cls") ;	Limpa a memória

Alterar cor do console

Para isso utilize o seguinte comando:

```
system ("color XX");
```

No lugar do x coloque o codigo para o fundo e para letra. **system ("color <FUNDO><LETRA>");**

Codigo	Cor
0	Preto
1	Azul
2	Verde
3	Verde-água
4	Vermelho
5	Roxo
6	Amarelo
7	Branco
8	Cinza
9	Azul claro
A	Verde claro
B	Verde-água claro
C	Vermelho claro
D	Lilás
E	Amarelo claro
F	Branco brilhante