
Programação – Anexo para Exames

Biblioteca <stdlib.h>

<pre>void *malloc(size_t size);</pre>
Requisita um espaço de memória dinâmica com (<i>size</i>) bytes. Devolve um ponteiro para o início do espaço alocado ou NULL em caso de erro.
<pre>void *realloc(void *ptr, size_t size);</pre>
Altera o número de bytes do espaço dinâmico referenciado por (<i>ptr</i>) para passar a ter (<i>size</i>) bytes. Devolve um ponteiro para o início do espaço com o novo tamanho ou NULL em caso de erro.
<pre>void free(void *ptr);</pre>
Liberta o espaço dinâmico referenciado por (<i>ptr</i>).
<pre>int atoi(const char *st);</pre>
Converte a string referenciada por (<i>st</i>) num inteiro. Devolve o valor inteiro obtido ou 0 caso a conversão seja impossível.
<pre>double atof(const char *st);</pre>
Converte a string referenciada por (<i>st</i>) num real. Devolve o valor real obtido ou 0 caso a conversão seja impossível.
<pre>int abs (int x);</pre>
Calcula o valor absoluto de (<i>x</i>) e devolve o resultado.

Biblioteca <string.h>

<pre>char *strcpy(char *strDest, const char *strSource);</pre>
Copia a string referenciada por (<i>strSource</i>) para a string referenciada por (<i>strDest</i>). Devolve o endereço da string <i>StrDest</i> .
<pre>char *strcat(char *strDest, const char *strSource);</pre>
Adiciona a string referenciada por (<i>strSource</i>) ao final da string referenciada por (<i>strDest</i>). Devolve o endereço da string <i>StrDest</i> .
<pre>size_t strlen(const char *str);</pre>
Devolve o número de caracteres da string referenciada por (<i>str</i>).
<pre>int strcmp(const char *str1, const char *str2);</pre>
Compara lexicograficamente as strings referenciadas por (<i>str1</i>) e (<i>str2</i>). Devolve 0 se foram iguais, um valor menor de que 0 se (<i>str1</i>) for lexicograficamente menor do que (<i>str2</i>) ou um valor maior do que 0 se (<i>str1</i>) for lexicograficamente superior a (<i>str2</i>).

Biblioteca <ctype.h>

<pre>int isalpha(int c);</pre>	<pre>int tolower(int c);</pre>
<pre>int isdigit(int c);</pre>	Recebe uma letra maiúscula (<i>c</i>) e devolve a correspondente minúscula.
<pre>int islower(int c);</pre>	<pre>int toupper(int c);</pre>
<pre>int isupper(int c);</pre>	Recebe uma letra minúscula (<i>c</i>) e devolve a correspondente maiúscula.
Verificam se o argumento recebido é do tipo desejado. Devolvem um valor diferente de 0 se isso suceder, ou 0 caso contrário.	

Biblioteca <stdio.h>

<pre>FILE *fopen(const char *stream, const char *mode);</pre>
Abre o ficheiro (<i>stream</i>) no modo de abertura especificado em (<i>mode</i>). Devolve um ponteiro para o FILE associado ao ficheiro que foi aberto ou NULL em caso de erro.
<pre>int fclose(FILE *stream);</pre>
Fecha o ficheiro (<i>stream</i>). Devolve 0 em caso de sucesso ou EOF se ocorrer um erro.
<pre>int fseek(FILE *stream, long offset, int origin);</pre>
Desloca o ponto de acesso do ficheiro (<i>stream</i>). O novo local é obtido a partir de um ponto origem (<i>origin</i> - SEEK_SET, SEEK_CUR, SEEK_END) e deslocando o número de bytes indicado em (<i>offset</i>). Devolve 0 em caso de sucesso ou um valor diferente de 0 se ocorrer um erro.
<pre>long ftell(FILE *stream);</pre>
Calcula o atual ponto de acesso do ficheiro (<i>stream</i>). Devolve o número de bytes entre o início e a posição corrente do ficheiro ou -1 em caso de erro.
<pre>size_t fread(void *buffer, size_t size, size_t n, FILE *stream);</pre>
Lê (<i>n</i>) objetos, cada um com número de bytes (<i>size</i>), do ficheiro (<i>stream</i>) para o endereço referenciado por (<i>buffer</i>). Devolve o número de objetos efetivamente lidos.
<pre>size_t fwrite(void *buffer, size_t size, size_t n, FILE *stream);</pre>
Escreve (<i>n</i>) objetos, cada um com número de bytes (<i>size</i>), do endereço referenciado por (<i>buffer</i>) para o ficheiro (<i>stream</i>). Devolve o número de objetos efetivamente escritos.
<pre>char *fgets(char *string, int n, FILE *stream);</pre>
Lê uma string com tamanho máximo (<i>n</i>) do ficheiro (<i>stream</i>) para o vetor referenciado por (<i>string</i>). Devolve o endereço do vetor onde foi colocada a informação ou NULL se fim de ficheiro.
<pre>int fputs(const char *string, FILE *stream);</pre>
Escreve a string referenciada por (<i>string</i>) para o ficheiro (<i>stream</i>). Devolve um valor não negativo ou EOF em caso de erro.
<pre>int fgetc(FILE *stream);</pre>
Lê o próximo carácter do ficheiro (<i>stream</i>). Devolve carácter lido ou EOF se fim de ficheiro.
<pre>int fputc(int c, FILE *stream);</pre>
Escreve o carácter (<i>c</i>) para o ficheiro (<i>stream</i>). Devolve o carácter escrito ou EOF em caso de erro.
<pre>int fscanf(FILE *stream, const char *format , [argument]...);</pre>
Lê informação do ficheiro (<i>stream</i>) para os endereços referenciados por [<i>argument</i>], de acordo com a sequência de formatação (<i>format</i>). Devolve o número de campos lidos e convertidos com sucesso ou EOF se fim de ficheiro
<pre>int fprintf(FILE *stream, const char *format , [argument]...);</pre>
Escreve as variáveis [<i>argument</i>] para o ficheiro (<i>stream</i>), de acordo com a sequência de formatação (<i>format</i>). Devolve o n.º de caracteres escritos ou um valor negativo em caso de erro.
<pre>int rename(const char *oldname, const char *newname);</pre>
Altera o nome atual do ficheiro (<i>oldname</i>) para o novo nome (<i>newname</i>). Devolve 0 em caso de sucesso ou um valor diferente de zero se ocorrer um erro
<pre>int remove(const char *name);</pre>
Elimina o ficheiro (<i>name</i>). Devolve 0 em caso de sucesso ou um valor diferente de zero se ocorrer um erro.