

ADPF





stdio.h





stdio.h - This is standard input/output header file in which Input/Output functions are declared

Function	Description
printf()	This function is used to print the character, string, float, integer, octal and hexadecimal values onto the output screen
scanf()	This function is used to read a character, string, numeric data from keyboard.
getc()	It reads character from file
gets()	It reads line from keyboard
getchar()	It reads character from keyboard
puts()	It writes line to o/p screen
putchar()	It writes a character to screen
clearerr()	This function clears the error indicators



stdio.h - This is standard input/output header file in which Input/Output functions are declared

Function	Description
f open()	All file handling functions are defined in stdio.h header file
f close()	closes an opened file
getw()	reads an integer from file
putw()	writes an integer to file
f getc()	reads a character from file
putc()	writes a character to file
f putc()	writes a character to file
f gets()	reads string from a file, one line at a time



stdio.h - This is standard input/output header file in which Input/Output functions are declared

Function	Description
f puts()	writes string to a file
f eof()	finds end of file
f getchar	reads a character from keyboard
f getc()	reads a character from file
f printf()	writes formatted data to a file
f scanf()	reads formatted data from a file
f getchar	reads a character from keyboard
f putchar	writes a character from keyboard



stdio.h - This is standard input/output header file in which Input/Output functions are declared

Function	Description
f seek()	moves file pointer position to given location
SEEK_SET	moves file pointer position to the beginning of the file
SEEK_CUR	moves file pointer position to given location
SEEK_END	moves file pointer position to the end of file.
f tell()	gives current position of file pointer
rewind()	moves file pointer position to the beginning of the file
putc()	writes a character to file
sprint()	writes formatted output to string



stdio.h - This is standard input/output header file in which Input/Output functions are declared

Function	Description
sscanf()	Reads formatted input from a string
remove()	deletes a file
fflush()	flushes a file



stdlib.h



stdlib.h This header file contains general functions used in C programs

Function	Description
malloc()	This function is used to allocate space in memory during the execution of the program.
calloc()	This function is also like malloc () function. But calloc () initializes the allocated memory to zero. But, malloc() doesn't
realloc()	This function modifies the allocated memory size by malloc () and calloc () functions to new size
free()	This function frees the allocated memory by malloc (), calloc (), realloc () functions and returns the memory to the system.
abs()	This function returns the absolute value of an integer . The absolute value of a number is always positive. Only integer values are supported in C.
div()	This function performs division operation



stdlib.h This header file contains general functions used in C programs

Function	Description
abort()	It terminates the C program
exit()	This function terminates the program and does not return any value
system()	This function is used to execute commands outside the C program.
atoi()	Converts string to int
atol()	Converts string to long
atof()	Converts string to float
strtod()	Converts string to double
strtol()	Converts string to long



stdlib.h This header file contains general functions used in C programs

Function	Description
getenv()	This function gets the current value of the environment variable
setenv()	This function sets the value for environment variable
putenv()	This function modifies the value for environment variable
perror()	This function displays most recent error that happened during library function call.
rand()	This function returns the random integer numbers
delay()	This function Suspends the execution of the program for particular time



string.h



string.h All string related functions are defined in this header file

String functions	Description
strcat ()	Concatenates str2 at the end of str1
strncat ()	Appends a portion of string to another
strcpy ()	Copies str2 into str1
strncpy ()	Copies given number of characters of one string to another
strlen ()	Gives the length of str1
strcmp ()	Returns 0 if str1 is same as str2. Returns <0 if str1 < str2. Returns >0 if str1 > str2
strncmpi ()	Same as strcmp() function. But, this function negotiates case. “A” and “a” are treated as same.
strchr ()	Returns pointer to first occurrence of char in str1



string.h All string related functions are defined in this header file

String functions	Description
strchr ()	last occurrence of given character in a string is found
strstr ()	Returns pointer to first occurrence of str2 in str1
strrstr ()	Returns pointer to last occurrence of str2 in str1
strdup ()	Duplicates the string
strlwr ()	Converts string to lowercase
strupr ()	Converts string to uppercase
strrev ()	Reverses the given string
strset ()	Sets all character in a string to given character

string.h All string related functions are defined in this header file

String functions	Description
strnset ()	It sets the portion of characters in a string to given character
strtok ()	Tokenizing given string using delimiter
memset()	It is used to initialize a specified number of bytes to null or any other value in the buffer
memcpy()	It is used to copy a specified number of bytes from one memory to another
memmove()	It is used to copy a specified number of bytes from one memory to another or to overlap on same memory.
memcmp()	It is used to compare specified number of characters from two buffers
memicmp()	It is used to compare specified number of characters from two buffers regardless of the case of the characters
memchr()	It is used to locate the first occurrence of the character in the specified string



math.h





math.h All maths related functions are defined in this header file

Function	Description
floor ()	This function returns the nearest integer which is less than or equal to the argument passed to this function.
round ()	This function returns the nearest integer value of the float/double/long double argument passed to this function. If decimal value is from “.1 to .5”, it returns integer value less than the argument. If decimal value is from “.6 to .9”, it returns the integer value greater than the argument.
ceil ()	This function returns nearest integer value which is greater than or equal to the argument passed to this function.
sin ()	This function is used to calculate sine value.
cos ()	This function is used to calculate cosine.
cosh ()	This function is used to calculate hyperbolic cosine.

math.h All maths related functions are defined in this header file

Function	Description
<code>exp ()</code>	This function is used to calculate the exponential “e” to the xth power.
<code>tan ()</code>	This function is used to calculate tangent.
<code>tanh ()</code>	This function is used to calculate hyperbolic tangent.
<code>sinh ()</code>	This function is used to calculate hyperbolic sine.
<code>log ()</code>	This function is used to calculates natural logarithm.
<code>log10 ()</code>	This function is used to calculates base 10 logarithm.
<code>sqrt ()</code>	This function is used to find square root of the argument passed to this function.
<u>pow ()</u>	This is used to find the power of the given number.
<code>trunc.(.)</code>	This function truncates the decimal value from floating point value and returns integer value.



ctype.h





ctype.h All character handling functions are defined in this header file

Functions	Description
isalpha()	checks whether character is alphabetic
isdigit()	checks whether character is digit
isalnum()	Checks whether character is alphanumeric
isspace()	Checks whether character is space
islower()	Checks whether character is lower case
isupper()	Checks whether character is upper case
isxdigit()	Checks whether character is hexadecimal
isctrl()	Checks whether character is a control character



cctype.h All character handling functions are defined in this header file

Functions	Description
isprint()	Checks whether character is a printable character
ispunct()	Checks whether character is a punctuation
isgraph()	Checks whether character is a graphical character
tolower()	Checks whether character is alphabetic & converts to lower case
toupper()	Checks whether character is alphabetic & converts to upper case



time.h



time.h This header file contains time and clock related functions

Functions	Description
setdate()	This function used to modify the system date
getdate()	This function is used to get the CPU time
clock()	This function is used to get current system time
time()	This function is used to get current system time as structure
difftime()	This function is used to get the difference between two given times
strftime()	This function is used to modify the actual time format
mktime()	This function interprets tm structure as calendar time
localtime()	This function shares the tm structure that contains date and time informations



time.h This header file contains time and clock related functions

Functions	Description
gmtime()	This function shares the tm structure that contains date and time informations
ctime()	This function is used to return string that contains date and time informations
asctime()	Tm structure contents are interpreted by this function as calendar time. This time is converted into string.



Other Functions - (Misc)

Header files	Functions declared in the header file
stdarg.h	va_start() - This function indicates the start process of variable length argument list in a program
	va_arg() - This function is used to fetch the arguments from variable length argument list
	va_end() - This function indicates the end process of variable length argument list in a program
signal.h	signal() - It is used to install signal handler
	raise() - It is used to raise signal in a C program
setjmp.h	setjmp() - This function prepares to use longjmp() function
	longjmp() - It is used for non local jump
locale.h	setlocale() - It sets locale()
	localeconv() - It gets locale conventions



Header files	Functions declared in the header file
errno.h	errno() - This function sets errno value to 0 at the beginning of the program. This value is modified to other than 0 when an error occurs while any function call.
assert.h	assert() - This function gets an integer as parameter. If this parameter is 0, writes message to stderr. Then, terminates the program. If this parameter is non 0, it does nothing.