

Programming cheatsheet

headers

math.h/ stdio.h/ ctype.h/ stdlib.h/ string.h/

isdigit, isgraph, isupper, toupper, tolower

strtof, strtoll, strtoul, strtoull, atoi()

strtod (const char* str, char** endptr)

strtol (const char* str, char** endptr, int base); $0 \leq \text{base} \leq 36$; if base==0, it will detect input's prefix automatically

strlen, strchr, strrchr, strcpy, strncpy, strstr

strncpy(char* dest, const char* source, size_t num);

strncat(char* dest, const char* source, size_t num);

strncmp(const char* str1, const char* str2, size_t num);

strchr(const char str, int character); *find the index of first 'character'*

strrchr(const char str, int character); *find the index of last 'character'*

strstr(const char str1, const char str2); *find the index of first substring "str2"*

strpbrk(const char str1, const char str2); *find the pointer of first character in the character set "str2"*

strspn(const char str1, const char str2); *count the length of string which consist of character set "str2"*

strcspn(const char* str1, const char* str2); *count the length of string which doesn't consist of character set "str2"*

strtok(char* token, const char* delimiters); In first call of strtok, *token have to be the address of string. After first call, token parameter have to be filled with NULL. Additionally, strtok is destructive, which means that it will set the character in the delimiters '\0'.

memset(void ptr, int value, size_t num); set [num] bytes after [ptr] [value]

etc

formatted output: %[flags][width][.precision][length]specifier

conversion specifier:

d: decimal integer/ u: unsigned decimal integer/ o: unsigned octal integer/ x: unsigned hexadecimal integer/ f: float/ e: float in scientific notation/ p: pointer

flags:

-: fill spaces at right till length of width/ 0: fill 0 at left till length of width/ +: insert + if the number is positive/ space: Print a space before a positive value/ #: prefix 0 or 0x before a octal or hexadecimal value

width: minimal number of character printed

if the length of number is longer than width, fill spaces at left. Else, the value will not be cut off.

formatted input: %[*][width][length]specifier

[]

%[[^]aeiou] string not consist of 'a', 'e', 'i', 'o', 'u'

%[[^]-+/%] *string exclusive of* '-', '+', ',', '/', '%'. It's noted that if '-' is specified, it has to be declare at first.

width: maximum number of characters be read

***: read but not stored**