|  |  |  |
| --- | --- | --- |
| ***Excerpt from ASCII table*** | | |
| *Sign* | *Description* | *Decimal value* |
| **LF** | *new line* | ***10*** |
| **Space** | *space, blank* | ***32*** |
| **0** | digit zero | ***48*** |
| **A** | uppercase letter A | ***65*** |
| **a** | lowercase letter a | ***97*** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| ***Priority and associativity of operators*** | | |
|  | *Operator* | *Operator*  *Associativity* |
| ←Hihher priority Lower priority→ | *function call***()**  **[] . ->**  *postfix* **++ --** | L → R |
| **! ~ sizeof**  *address***&** *indirection***\***  *prefix* **++ --**  *unarni* **+ -** | R → L |
| **(cast)** | R → L |
| *arithmetical* **\* / %** | L → R |
| *binary* **+ -** | L → R |
| **<< >>** | L → R |
| **< <= > >=** | L → R |
| **== !=** | L → R |
| *bitwise*  **&** | L → R |
| **^** | L → R |
| **|** | L → R |
| **&&** | L → R |
| **||** | L → R |
| **? :** | R → L |
| **= \*= /= %= += -=** **&= ^= |= <<= >>=** | R → L |
| *operator* **,** | L → R |

|  |  |
| --- | --- |
| ***Real number representation*** | |
| *IEEE 754 - 32 bita* | *IEEE 754 - 64 bita* |
| K = BE + 127 | K = BE + 1023 |
| denormalized: K = 0 | denormalized: K = 0 |
| ± ∞ ili NaN: K = 255 | ± ∞ili NaN: K = 2047 |
| largest normalized   3.4 × 1038 | largest normalized  1.8 × 10308 |
| smallest positive norm.   1.17 × 10-38 | smallest positive norm.  2.2 × 10-308 |
| 2-24  6 × 10-8 | 2-53  1.1 × 10-16 |

**Prefixes for integer constants**

0 : *oktalno*

0x*,* 0X : *heksadekadski*

**Suffixes for integer constants**

L*,* l : *long*

|  |  |
| --- | --- |
| ***Memory usage (gcc, x86\_64)*** | |
| Tip | Broj bajtova |
| char | 1 |
| short | 2 |
| int | 4 |
| long | 4 |
| long long | 8 |
| float | 4 |
| double | 8 |
| long double | 12 |

LL*,* ll : *long long*

|  |
| --- |
| ***gcc options*** |
| -E |
| -std=c11 |
| -Wall |
| -pedantic-errors |
| -S |
| -c |
| -o |
| -save-temps |
| --verbose |

U*,* u *: unsigned*

**Suffixes for real constants**

F, f : *float*

L*,* l : *long double*

**Standard library**

***stdlib.h***

size\_t

EXIT\_FAILURE, EXIT\_SUCCESS

RAND\_MAX, NULL

int abs(int x); |x|

long labs(long x);

long long llabs(long long x);

void srand(unsigned int seed);

int rand(void);

void exit(int status);

void \*malloc(size\_t size);

void \*realloc(void \*ptr, size\_t size);

void free(void \*ptr);

***math.h***

double fabs(double x); |x|

float fabsf(float x);

long double fabsl(long double x);

double pow(double x, double y); xy

double sqrt(double x); 

double exp(double x); ex

double log(double x); ln x

double log10(double x); log10 x

double sin(double x);

double cos(double x);

double tan(double x);

double asin(double x);

double acos(double x);

double atan(double x);

double atan2(double y, double x);

double cosh(double x);

double sinh(double x);

double tanh(double x);

double ceil(double x); ⌈x⌉

double floor(double x); ⌊x⌋

double fmod(double x, double y);

|  |  |
| --- | --- |
| ***Keywords*** | |
| auto | short |
| break | signed |
| case | sizeof |
| char | static |
| const | struct |
| continue | switch |
| default | typedef |
| do | union |
| double | unsigned |
| else | void |
| enum | volatile |
| extern | while |
| float | \_Alignas |
| for | \_Alignof |
| goto | \_Atomic |
| if | \_Bool |
| inline | \_Complex |
| int | \_Generic |
| long | \_Imaginary |
| register | \_Noreturn |
| restrict | \_Static\_assert |
| return | \_Thread\_local |

***time.h***

size\_t, time\_t

NULL

time\_t time(time\_t \*timer);

***string.h***

size\_t

NULL

char \*strcpy(char \*s1, const char \*s2);

char \*strncpy(char \*s1, const char \*s2, size\_t n);

char \*strcat(char \*s1, const char \*s2);

char \*strncat(char \*s1, const char \*s2, size\_t n);

size\_t \*strlen(const char \*s);

int \*strcmp(const char \*s1, const char \*s2); s1 < s2 ⇒ strcmp < 0

int \*strncmp(const char \*s1, const char \*s2, size\_t n);

char \*strchr(const char \*s, int c); NULL if not found

char \*strrchr(const char \*s, int c); NULL if not found

char \*strstr(const char \*s1, const char \*s2); NULL if not found

char \*strpbrk(const char \*s1, const char \*s2); NULL if not found

***ctype.h***

int isdigit(int c);

int isdigit(int c);

int isxdigit(int c);

int isalpha(int c);

int isalnum(int c);

int isprint(int c);

int iscntrl(int c);

int isspace(int c);

int islower(int c);

int isupper(int c);

int toupper(int c);

int tolower(int c);

***stdio.h***

stdin, stdout, stderr

size\_t, FILE

NULL, EOF, SEEK\_CUR, SEEK\_END, SEEK\_SET

int scanf(const char \*format, ...);

int sscanf(const char \*buffer, const char \*format, ...);

int fscanf(FILE \*stream, const char \*format, ...);

*general form of the conversion specification:* %[\*][width][modifier]specifier

*specifiers for scanf:* c, d, u, o, x, f, s, p, [znakovi], [^znakovi]

*modifiers for* d*,* u*,* o*,* x: h - short l - long ll - long long

*modifiers for* f: l - double L - long double

int printf(const char \*format, ...);

int sprintf(char \*buffer, const char \*format, ...);

int fprintf(FILE \*stream, const char \*format, ...);

*general form of the conversion specification:* %[character][width][.precision][modifier]specifier

*specifiers for printf:* c, d, u, o, x, X, f, e, E, g, G, s, p

*modifiers for* d*,* u*,* o*,* x: h - short l - long ll - long long

*modifiers for* f: l - double L - long double

int getc(FILE \*stream); int getchar(void); error, eof → EOF

int ungetc(int c, FILE \*stream); error → EOF

int putc(int c, FILE \*stream); int putchar(int c); error → EOF

char \*fgets(char \*s, int n, FILE \*stream); error, eof → NULL

int fputs(const char \*s, FILE \*stream); int puts(const char \*s);error → EOF

FILE \*fopen(const char \*filename, const char \*mode); error → NULL

*mode:* "w[b]","a[b]","r[b]","w+[b]","a+[b]","r+[b]"

int fflush(FILE \*stream); error → EOF

int fclose(FILE \*stream); error → EOF

size\_t fread(void \*ptr, size\_t size, size\_t n, FILE \*stream); error → <n

size\_t fwrite(const void \*ptr, size\_t size, size\_t n, FILE \*stream); error → <n

int fseek(FILE \*stream, long offset, int whence); error → int≠0

*whence:* SEEK\_SET relative to the beginning of the file

*SEEK\_CUR* in relation to the current position

*SEEK\_END* relative to the end of the file

long ftell(FILE \*stream); error → -1L