C CONTROL STRUCTURES

```
if ( … ) {
        …
}
else if ( … ) {
        …
}
else {
        …
}
```

```
switch (integer-expr){
case constant-label:
        …
        break;
…
default:
        …
        Break;
}
```

Operators	Associativity	Type
() [] ++ *(postfix)* -- *(postfix)*	left to right	postfix
+ - ++ -- ! * & (*type*)	right to left	unary
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > s>=	left to right	relational
== !=	left to right	equality
&&	left to right	logical AND
||	left to right	logical OR
?:	right to left	conditional
= += -= *= /= %=	right to left	assignment
,	left to right	comma

COMMON C FUNCTIONS

- sizeof()

stdio.h

- size_t
- EOF
- stdin
- putchar()
- printf()
- NULL
- FILE
- stdout
- getchar()
- scanf()
- int sprintf(char *s, const char *format, …) - returns the number of characters written to s
- int sscanf(char *s, const char *format, …) - returns the number of items successfully read from s
- FILE* fopen(const char *filename, const char *mode) - returns NULL when failed (r, w, a, r+, rb, wb, ab, rb+)
- fclose(FILE *fp)
- int feof(FILE *fp) - tests the end-of-file indicator for given stream
- fputc(int c, FILE *fp) - writes the character c to the stream
- int fgetc(FILE *fp) - gets the next character from the stream
- int fputs(char *buf, FILE *fp) - returns number of chars written to stream from buf
- char* fgets(char *buf, int n, FILE *fp) - reads the next line or (n-1) characters from the stream and stores it into buf, returns NULL if end-of-file
- int fprintf(FILE *fp, const char *format, ...) - returns the number of characters written to stream
- int fscanf(FILE *fp, const char *format, ...) - returns the number of items read from stream or EOF
- rewind(FILE *fp) - sets the file position to the beginning of the file
- size_t fwrite(void *ptr, size_t size, size_t nmemb, FILE *stream) - writes nmemb elements, each with size bytes, to the stream and returns the number of elements successfully written
- size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream) reads nmemb elements, each with size bytes, from the stream and returns the number of elements successfully read
- int fseek(FILE *stream, long int offset, int whence) - sets the file position to the offset, returns 0 if successful
  - offset is the number of bytes to seek from location whence (SEEK_SET, SEEK_CUR, SEEK_END)

string.h

- size_t
- strlen()
- strcpy(char *s1, const char *s2) - copies its second argument into its first argument
- strcat(char *s1, const char *s2) - appends its second argument to its first argument
- memmove(void *p1, const void *p2, size_t n) - copies n characters from p2 to p1
- memset(void *p, int c, size_t n) - sets the first n characters of p to the character c
- char* strtok(char *s, const char *delim) - returns NULL If no tokens remain