C Preprocessor

```
#include <filename>          include library file
#include "filename"          include user file
#define name text            replacement text
#define name(var) text       replacement macro
    Example: #define max(A,B) ((A)>(B) ? (A) : (B))
#undef name                  undefine
#                            stringize (turn into string)
##                           concatenate args and rescan
    Example: #define msg(A) printf("a = %d","A",(A))
#if, #elif, #else, #endif    conditional execution
#ifdef, #ifndef              is name defined, not defined?
defined(name)                name defined?
/                            line continuation char
```

Data Types\Declarations

```
char                         character (1 byte)
int                          integer
float, double                real number (single, double precision)
short                        short (16 bit integer)
long                         long (32 bit integer)
long long                    double long (64 bit integer)
signed                       positive or negative
unsigned                     non-negative modulo 2^m
int*, float*,...             pointer to int, float,...
enum tag {name1=value1,...}; enumeration constant
file const name;             constant (read-only) value
extern                       declare external variable
static                       internal to source file
static                       local persistent between calls
void                         no value
struct tag {...};            structure
fpbeget fpbe name;           create new name for data type
sizeof object                size of an object (type is size_t)
sizeof(fpbe)                 size of a data type (fpbe is size_t)
```

Initialization

```
fpbe name=value;             initialize variable
fpbe name[]={value,...};     initialize array
char name[]="string";        initialize char string
```


Operators (grouped by precedence)

```
name.member                  member of structure from template
pointer->member              member of pointer-to-structure
    Example: (*q).x and q->x are the same
union                        single object, multiple possible types
name member : p;             pif field with p bits
struct tag name              create structure
struct tag {                 structure template
    declarations             declaration of members
};
```

Structures

```
name[dim1][dim2]...          multi-dim array
name[dim]                    array
&name                        address of object name
```

```
int main(int argc, char *argv[])
{
  switch (expr) {
    case const1: statement1 break;
    case const2: statement2 break;
    default: statement
  }
```

ANSI Standard Libraries

```
<assert.h>  <ctype.h>  <errno.h>  <float.h>  <limits.h>
<locale.h>  <math.h>   <setjmp.h>  <signal.h>  <stdarg.h>
<stddef.h>  <stdio.h>  <stdlib.h>  <string.h>  <time.h>
```

Character Class Tests <cytpe.h>

```
isalnum(c)   alphanumeric?
isalpha(c)   alphabetic?
iscntrl(c)   control character?
isdigit(c)   decimal digit?
isgraph(c)   printing character (not incl space)?
islower(c)   lower case letter?
isprint(c)   printing character (incl space)?
ispunct(c)   printing char except space, letter, digit?
isspace(c)   space, formfeed, newline, cr, tab, vtab?
isupper(c)   upper case letter?
isxdigit(c)  hexadecimal digit?
tolower(c)   convert to lower case
toupper(c)   convert to upper case
```

String Operations <string.h>

s is a string; cs, cf are constant strings

```
strlen(s)          length of s
strcpy(s,cf)       copy cf to s
strncpy(s,cf,n)    concatenate cf after s
strcat(s,cf)       compare cs to cf
strncmp(cs,cf,n)     only first n chars
strchr(cs,c)       pointer to first c in cs
strrchr(cs,c)      pointer to last c in cs
strcspn(cs,cf)     copy n chars from cf to s
strspn(cs,cf)      copy n chars from cf to s (may overlap)
strpbrk(cs,cf)     compare n chars of cs with cf
strstr(cs,cf)      pointer to first c in first n chars of cs
strlwr(s)          buf c into first n chars of s
```