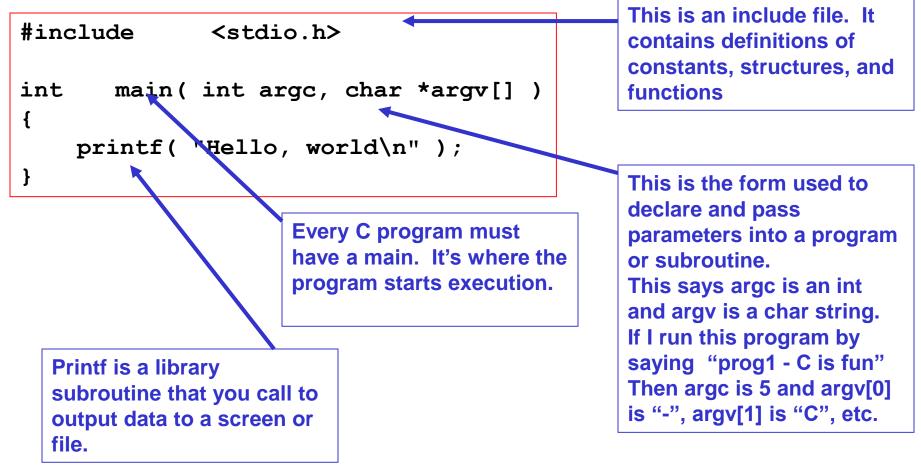


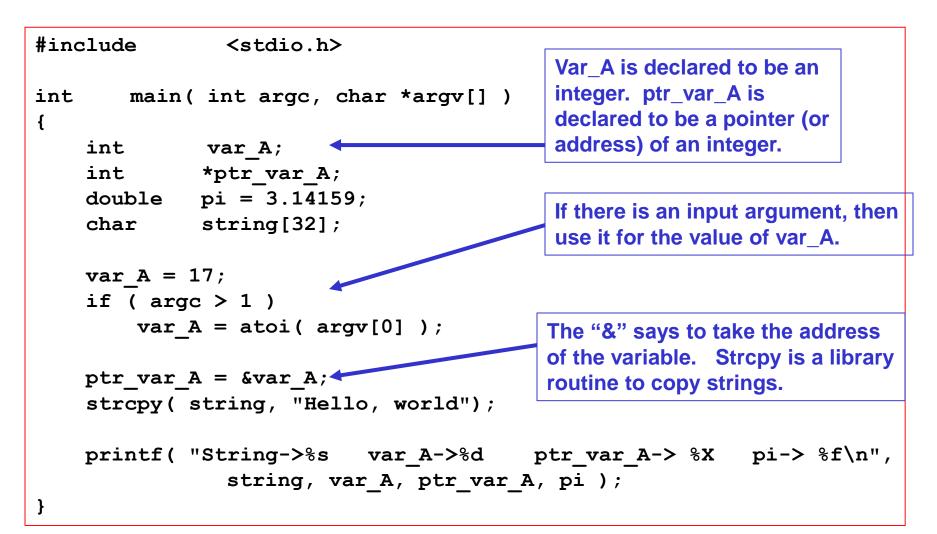
The assumption is that you know Java and need to extend that knowledge so you can program in C.

- 1. Hello world
- 2. declarations
- 3. pass by value/reference
- 4. arrays & structures
- 5. pointers
- 6. library functions
- 7. Compiling and Debugging gcc and gdb
- 8. Shell Commands

1. "Hello World" – The Simplest Program

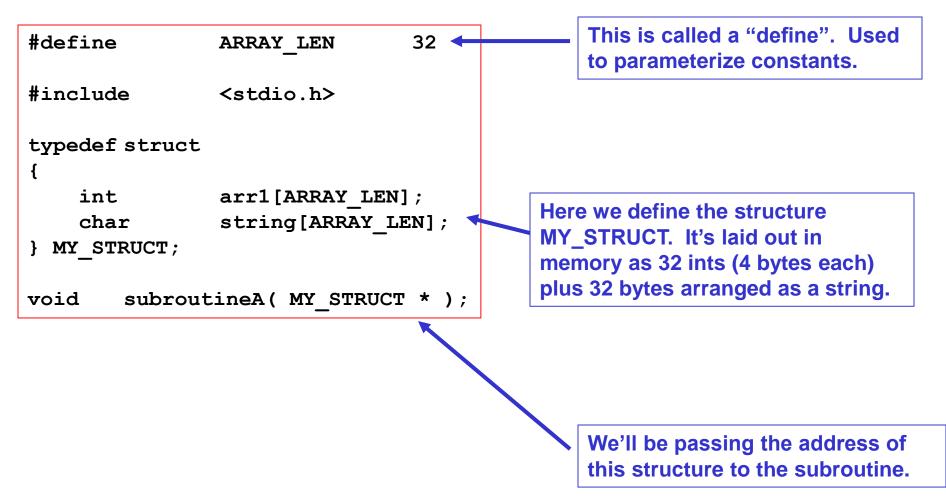


Declarations

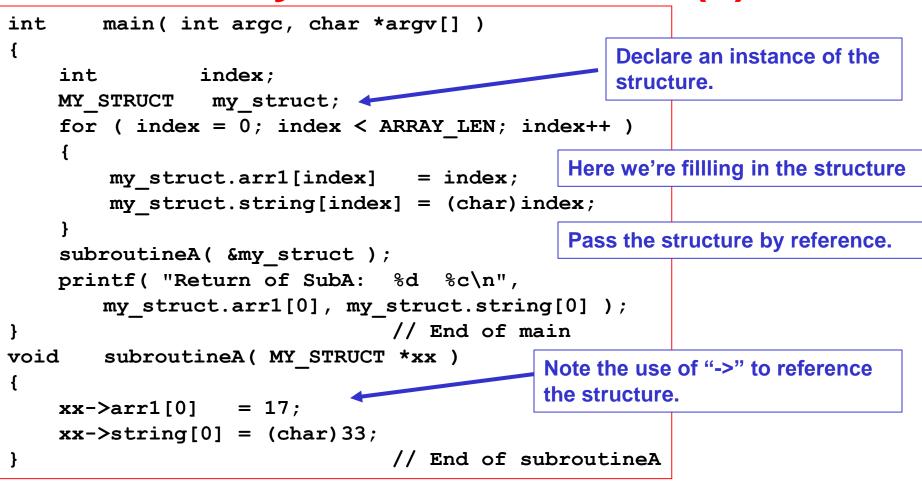


Pass by Value and Pass by Reference #include <stdio.h> These are called "prototypes". subroutineA(int , int *); void short functionB(int , int); int main(int argc, char *argv[]) { int var A; Var_A is passed by value. Note int var B; the & on var B which means that its address is passed in. var A = 17;**var** B = 33;subroutineA(var A, &var B); printf("Return from SubA: %d %d\n", var A, var B); printf("Return from FunB: %d\n", functionB(var A, var B)); // End of main } subroutineA(int A, int *B) void Note how int and ptr to int are { declared. *B = A; // B is a pointer to an int // End of subroutineA } short functionB(int A, int B) **{** return (A + B);4 functionB }

Arrays and Structures (1)

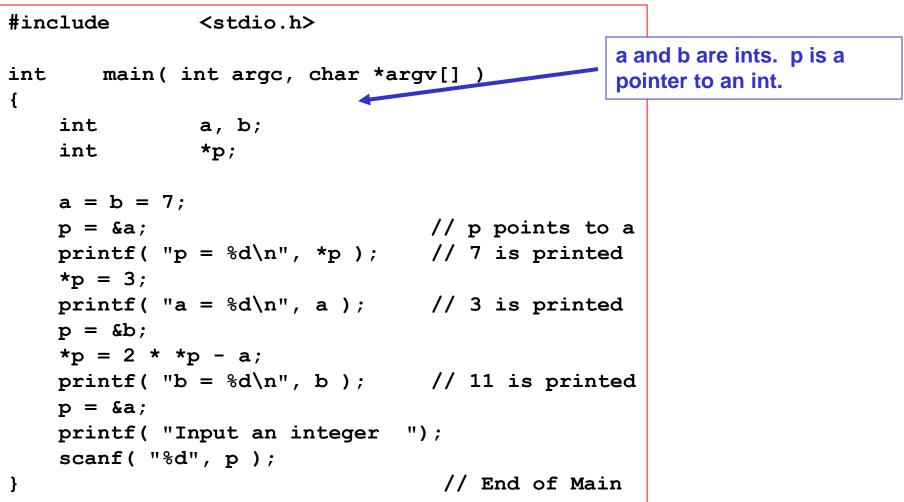


Arrays and Structures (2)



What is printed out by this program?

Pointers



Library Functions (1)

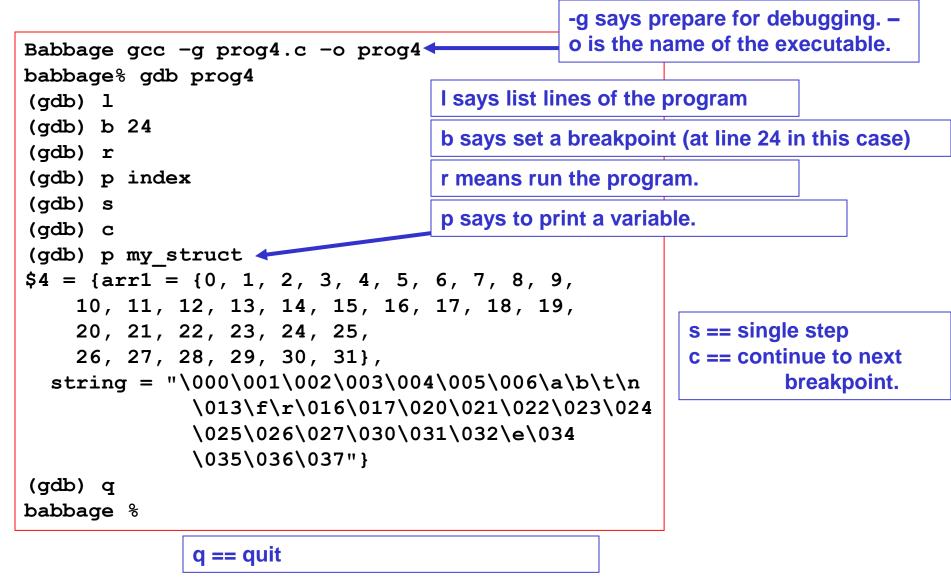
These pages show include files and the library function prototypes that are contained in them.

```
#include
             <ctype.h>
int isalnum(int c); // returns TRUE if char is alpha or numeric
int isspace(int c); // returns TRUE if char is white space
int tolower(int c); // returns the conversion of c to lower case
#include <math.h>
double cos(double x);
double exp(double x);
#include <stddef.h>
#typedef unsigned size t;
#define NULL ((void *) 0)
#define offsetof(s type, m) \
            ((size t) &(((s type *) 0) ->m ))
#include <stdio.h>
#define EOF (-1)
                          // End of File
#define NULL
             0
```

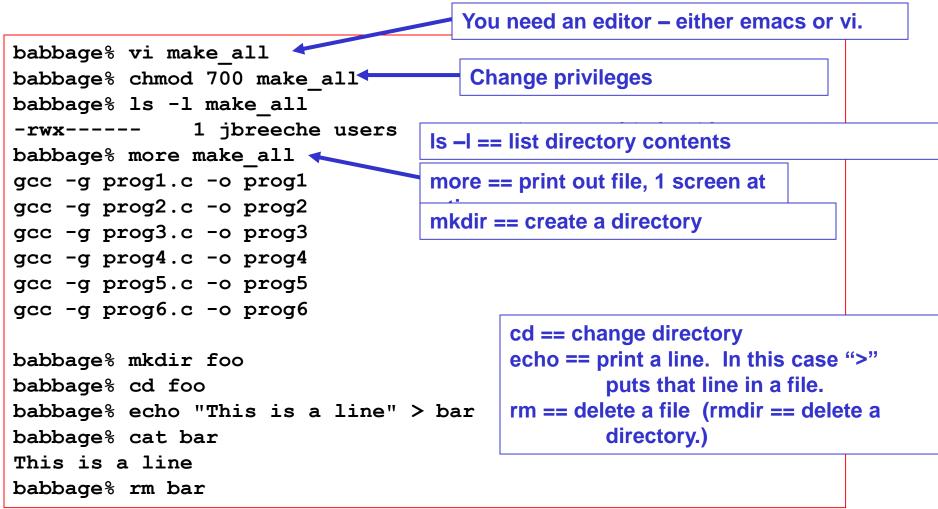
Library Functions (2)

```
#include <stdlib.h>
int atoi(const char *s);
int rand(void);
void *malloc(size_t size); // Allocates "size" bytes of memory
#include <string.h>
void *memcpy(void *to, void *from, size_t n);
char *strcat(char *s1, char *s2); // Concatenates
size t strlen(char *s);
```

Compilation And Debugging



Shell Commands



"man" and "apropos" are my favorites.