

Review of basic C programming ===== Basic
Structure ----- e.g. main() { printf("Hello World\n"); } Basic Data Type
(and printf conversion specification) -----
* long double (%Lf) * double (%f) * float (%f) * unsigned long int (%lu) * long int
(%ld) * unsigned int (%u) * int (%d) * short (%hd) * char (%c) * void Program
Control ----- * For loop e.g. for (i=0;i<100;i++) { printf("i=%d\n",i); } *
While loop e.g. i=0; while (i<100) { printf("i=%d\n",i); i++; } * Do-While loop e.g.
i=0; do { i++; printf("i=%d\n",i); } while (i<100); * If-Then-Else control e.g. if
(i>10) { printf("i is larger than 10\n"); } else { printf("i is smaller than or equal
to 10\n"); } * break and continue e.g. for (i=0;i<100;i++) { if (i==20) { break; }
printf("i=%d\n",i); } e.g. for (i=0;i<100;i++) { if (i==20) { continue; }
printf("i=%d\n",i); } * Switch-Case control e.g. switch (grade) { case 'A': printf("No
grade A\n"); break; case 'B': printf("grade B\n"); break; default: printf("No
good\n"); break; } Operators ----- * &&, || * () * ++, -- * +, -, *, /, % * ! *
(type) * <, <=, >, >= * ==, != * ?: * =, +=, -=, *=, /=, %= Functions ----- * math
functions (#include <math.h>) - sqrt(x) - exp(x) - log(x) - log10(x) -
fabs(x) - ceil(x) - floor(x) - pow(x,y) - fmod(x,y) - sin(x) - cos(x)
- tan(x) * function definitions and prototype e.g. int square(int y); main() {
b = square(a); } int square(int y) { return y*y; } * header files - #
include <...> - #include "..." * calling functions: call by value and call by reference e.g.
void changeit(int x); main() { int a; changeit(a); } void
changeit(int x) { x = 10; } e.g. void changeit(int *x); main() { int a;
changeit(&a); } void changeit(int *x) { *x = 10; } * local and global
scope void printa(); int a=5; main() { int a; a = 10; printa(); printf("a=%d\n",
a); } void printa() { printf("a=%d\n",a); a = 20; } * Recursion (non-iterative) e.
g. long factorial(long number); main() { factorial(10); } long factorial(long number)
; { if (number <= 1) return 1; else return (number * factorial(number - 1)); }
Arrays ----- * start from zero * declaration e.g. int c[5] = {3, 10, 22, 64, 23}; * as
a constant pointer - d = *(c+3); - d = c[3]; - e = c + 2; - e = &c[2]; * two
dimensional array: e.g. int c[10][5]; Pointers ----- * address in memory *
declaration e.g. int *countPtr, count; * operators e.g. countPtr = &count; *countPtr = 10;
Characters and Strings ----- * character array e.g. char color[] = "blue"
; * null terminate ('\0' or NULL) * conversion functions - atof - atoi - atol *
string manipulation functions - strcpy - strncpy - strcat - strncat - strcmp -
strncmp Input and Output ----- * getchar * gets * putchar * puts *
sprintf * sscanf * printf - for integer: %d, %i, %o, %u, %x - for floating-point: %e, %f,
%g - for character: %c, %s - field width and precision e.g. printf("%9.3f",123.
456789); Exercise ===== 6.15 Use a single-subscripted array to solve the
following problem. Read 20 numbers, each of which is between 10 and 100, inclusive. As each
number is read, print it only if it is not a duplicate of a number already read. Provide for
the "worst case" in which all 20 numbers are different. Use the smallest possible array to
solve this problem. 8.36 Dates are commonly printed in several different formats in
business correspondence. Two of the more common formats are: 07/21/55 and July 21,
1955 Write a program that reads a date in the first format and prints that date in the second
format.