main( ){

int a[100];

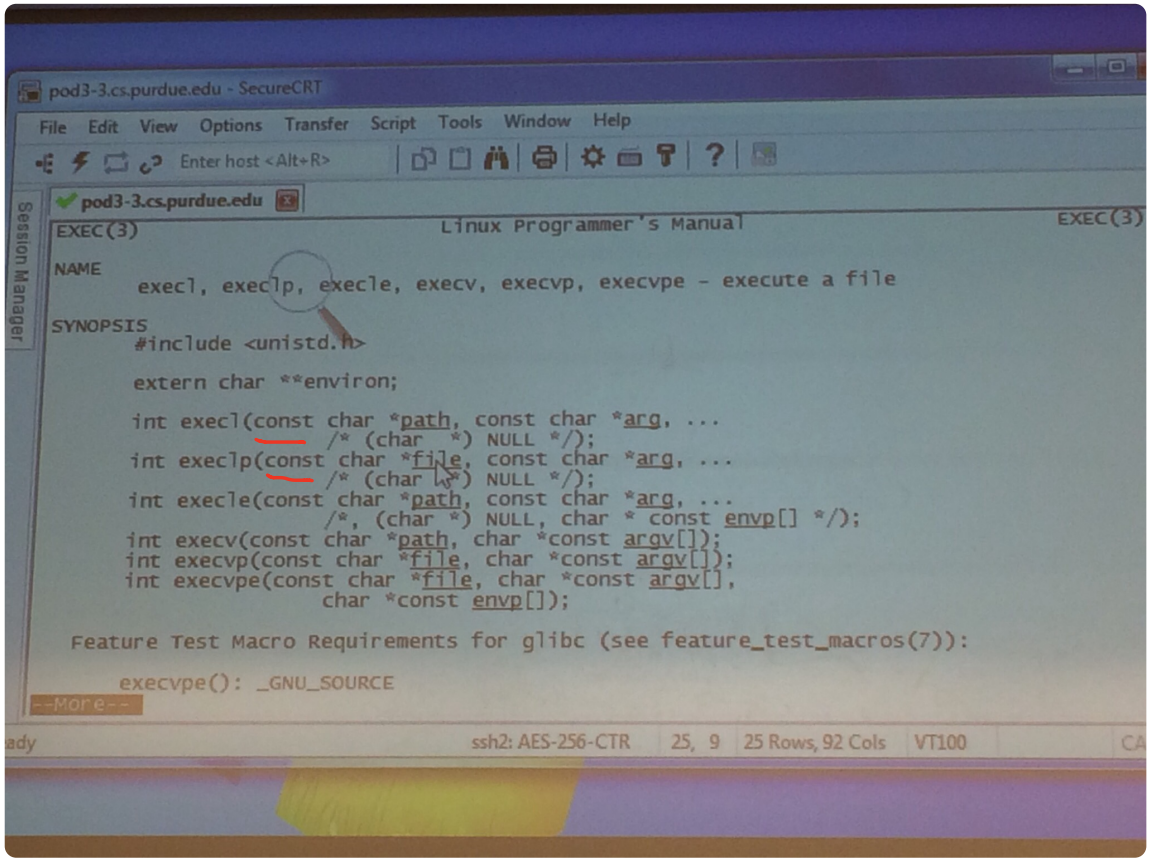
abc(a ,)

const int *

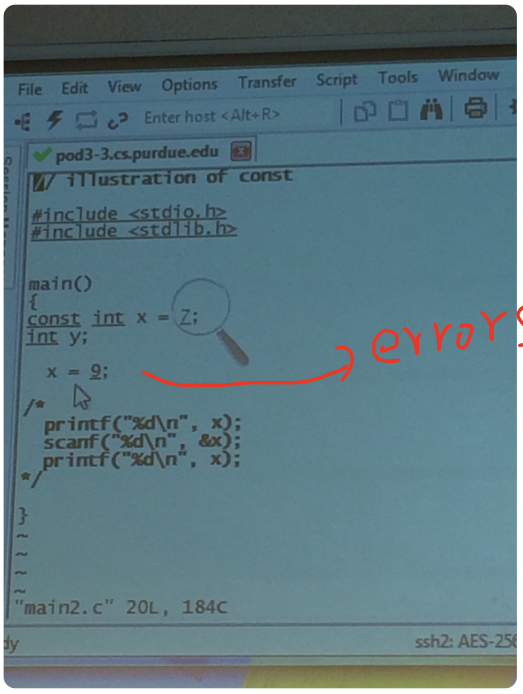
}

abc(const * x){

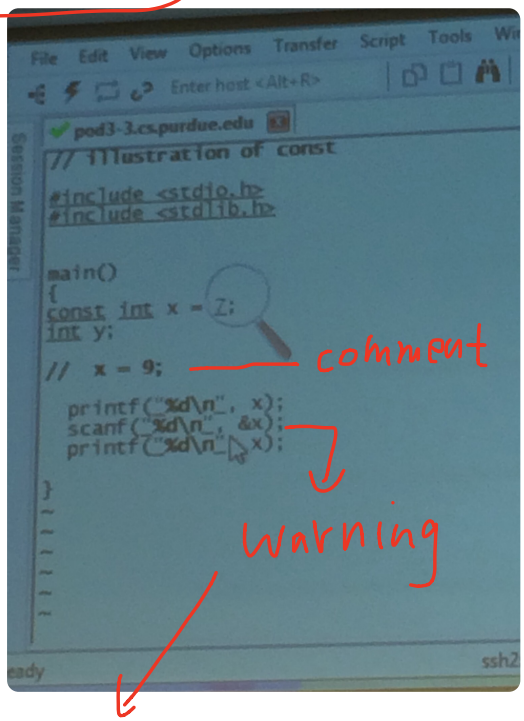
}



Const



errors



comment out

warning

But when running
it might cause
undeclared case,
Bugs might happen
might not happen
depends on compiler.

execvp ("ls" "ls" "-a" "-l" NULL)

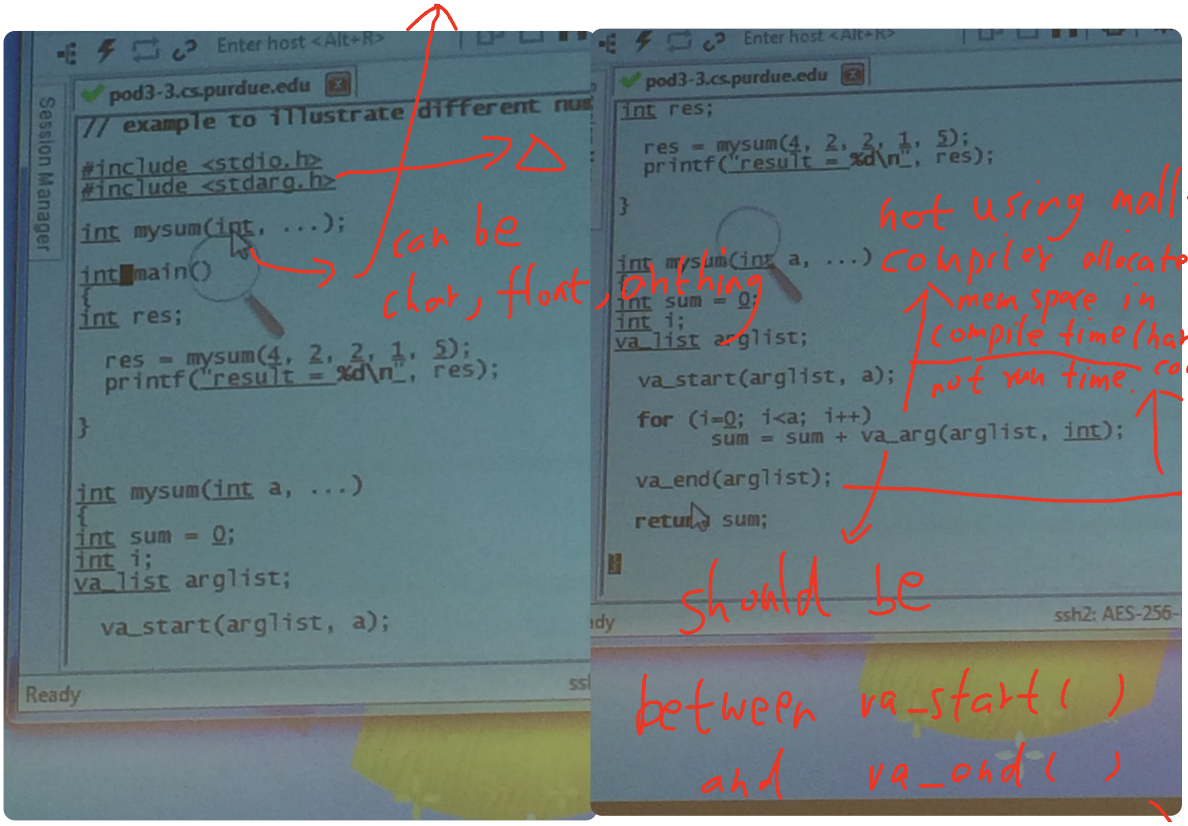
printf(" ", 0, 0)

mysum (2, 2, 2) = 6

mysum (7, 3) = 0

variable number of arguments

Must have one argument (at least)



calling va_arg first time return
the first argument after a.
which is 2 (the second argument)

calling ... second time return

... second ... after a

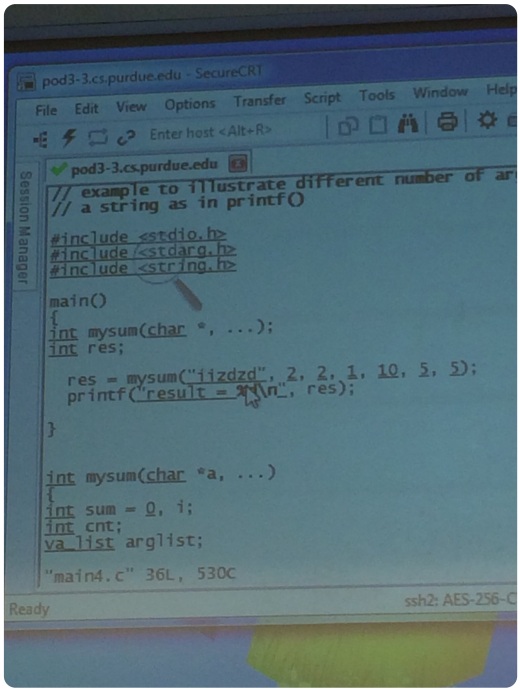
the third argument, which is 2.

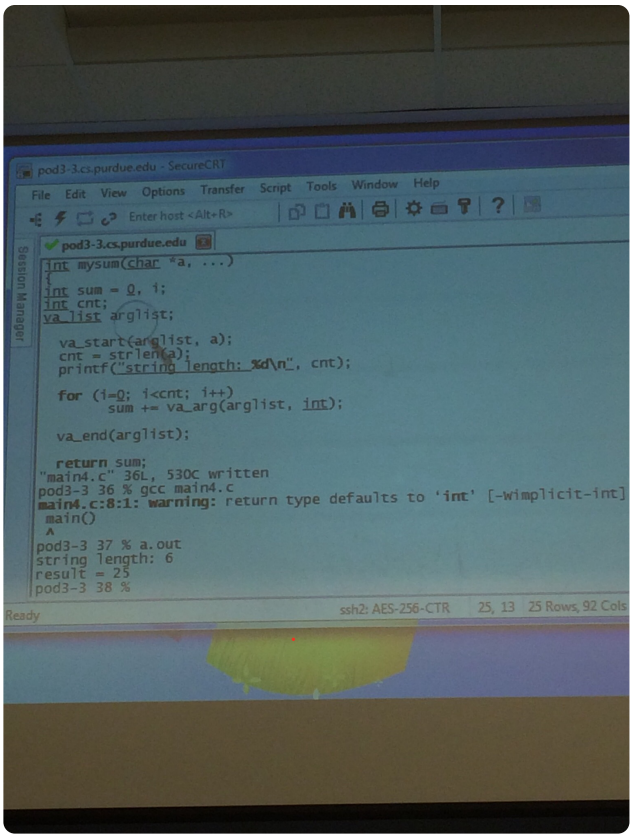
calll third time return

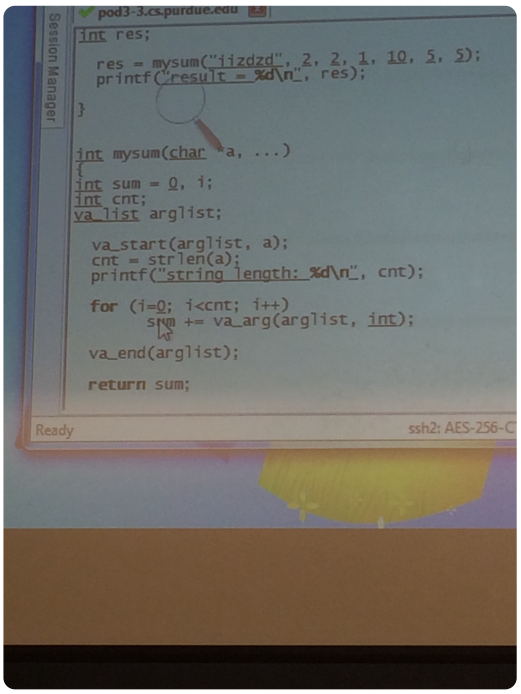
third ... after a

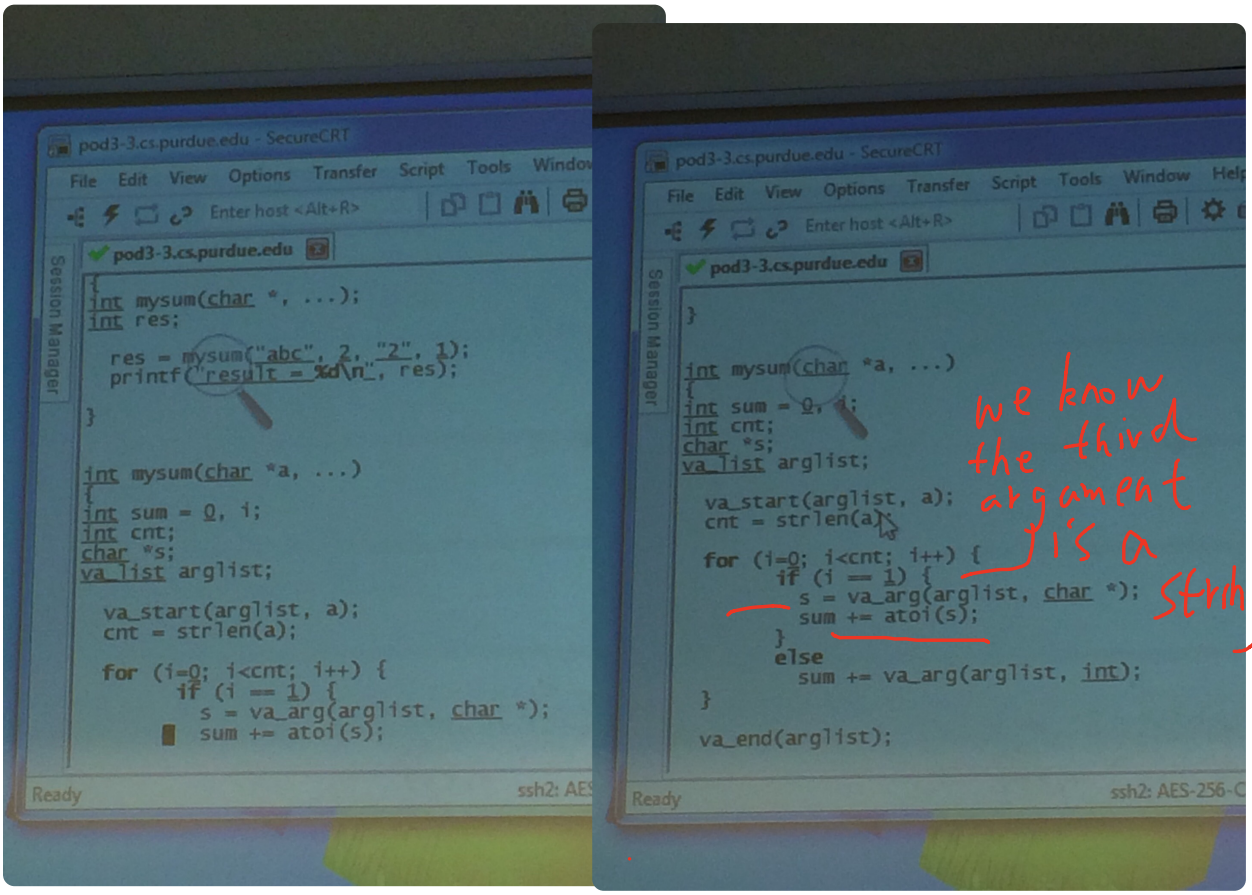
the fourth argumen which is 1.

....

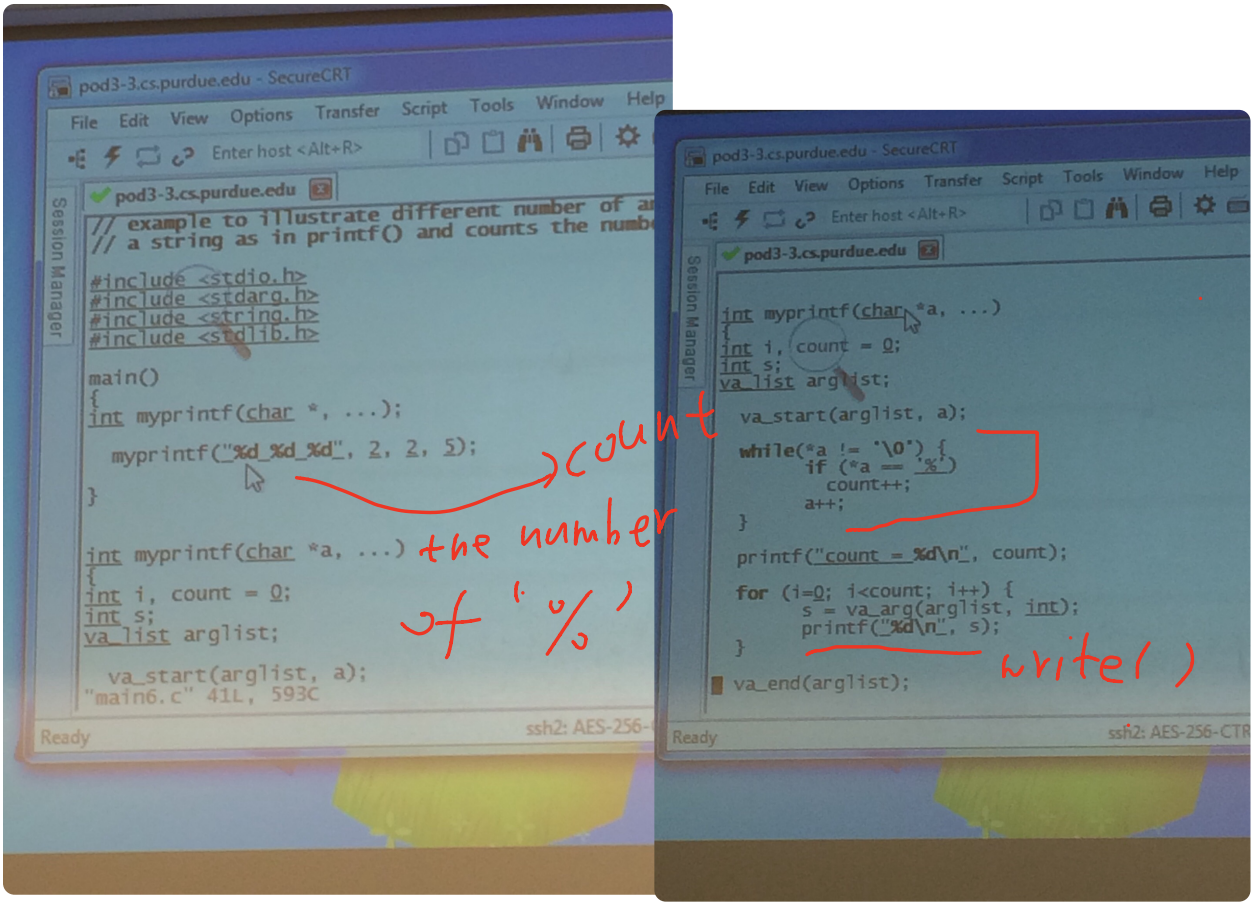

pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Session Manager
pod3-3.cs.purdue.edu
// example to illustrate different number of ar
// a string as in printf()
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
main()
{
int mysum(char *, ...);
int res;
res = mysum("iizdzd", 2, 2, 1, 10, 5, 5);
printf("result = %d\n", res);
}
int mysum(char *a, ...)
{
int sum = 0, i;
int cnt;
va_list arglist;
"main4.c" 36L, 530C
Ready
ssh2: AES-256-C

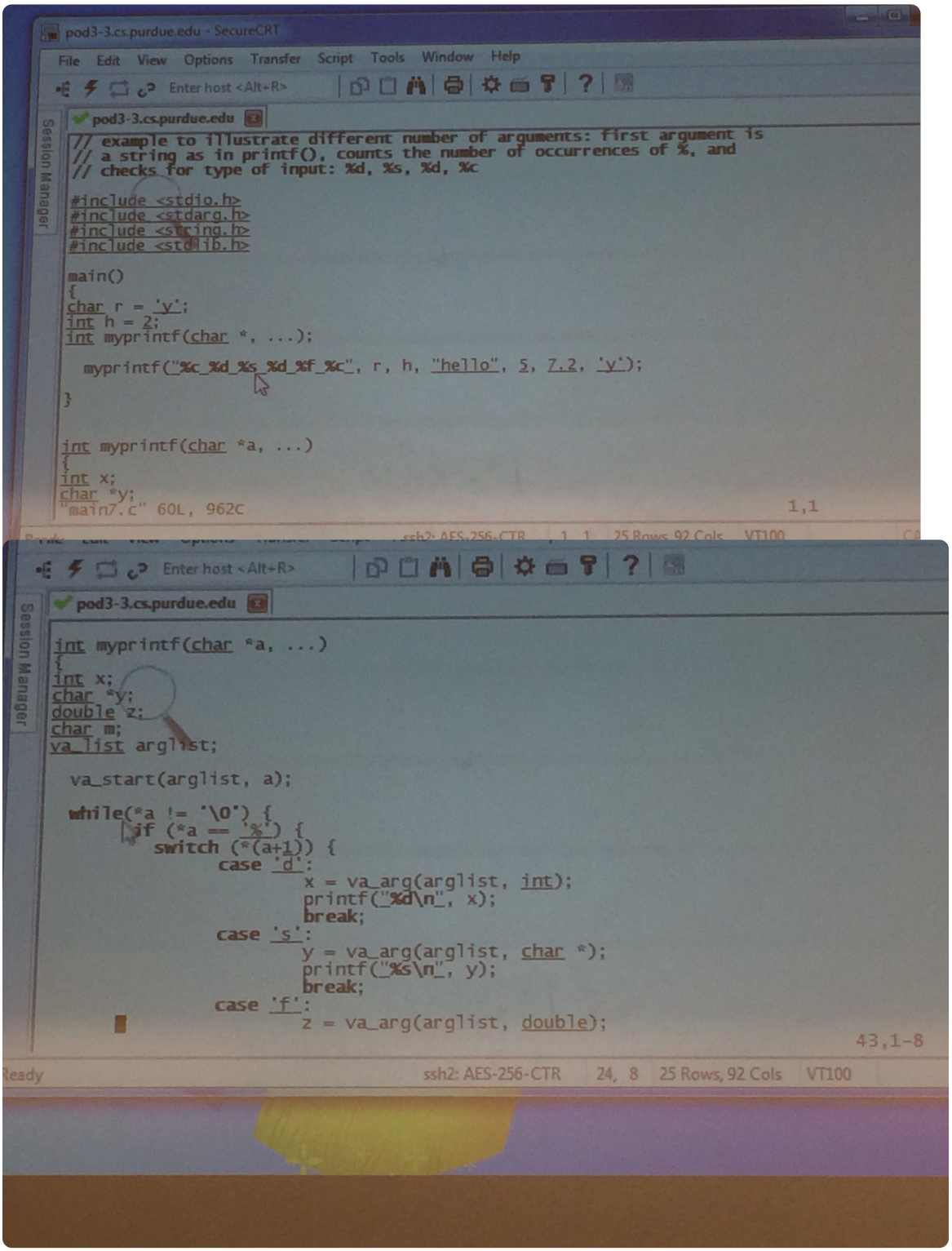

pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Session Manager
pod3-3.cs.purdue.edu
int mysum(char *a, ...)
{
int sum = 0, i;
int cnt;
va_list arglist;
va_start(arglist, a);
cnt = strlen(a);
printf("string length: %d\n", cnt);
for (i=0; i<cnt; i++)
sum += va_arg(arglist, int);
va_end(arglist);
return sum;
"main4.c" 36L, 530C written
pod3-3 36 % gcc main4.c
main4.c:8:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^
pod3-3 37 % a.out
string length: 6
result = 25
pod3-3 38 %
Ready
ssh2: AES-256-CTR
25, 13
25 Rows, 92 Cols

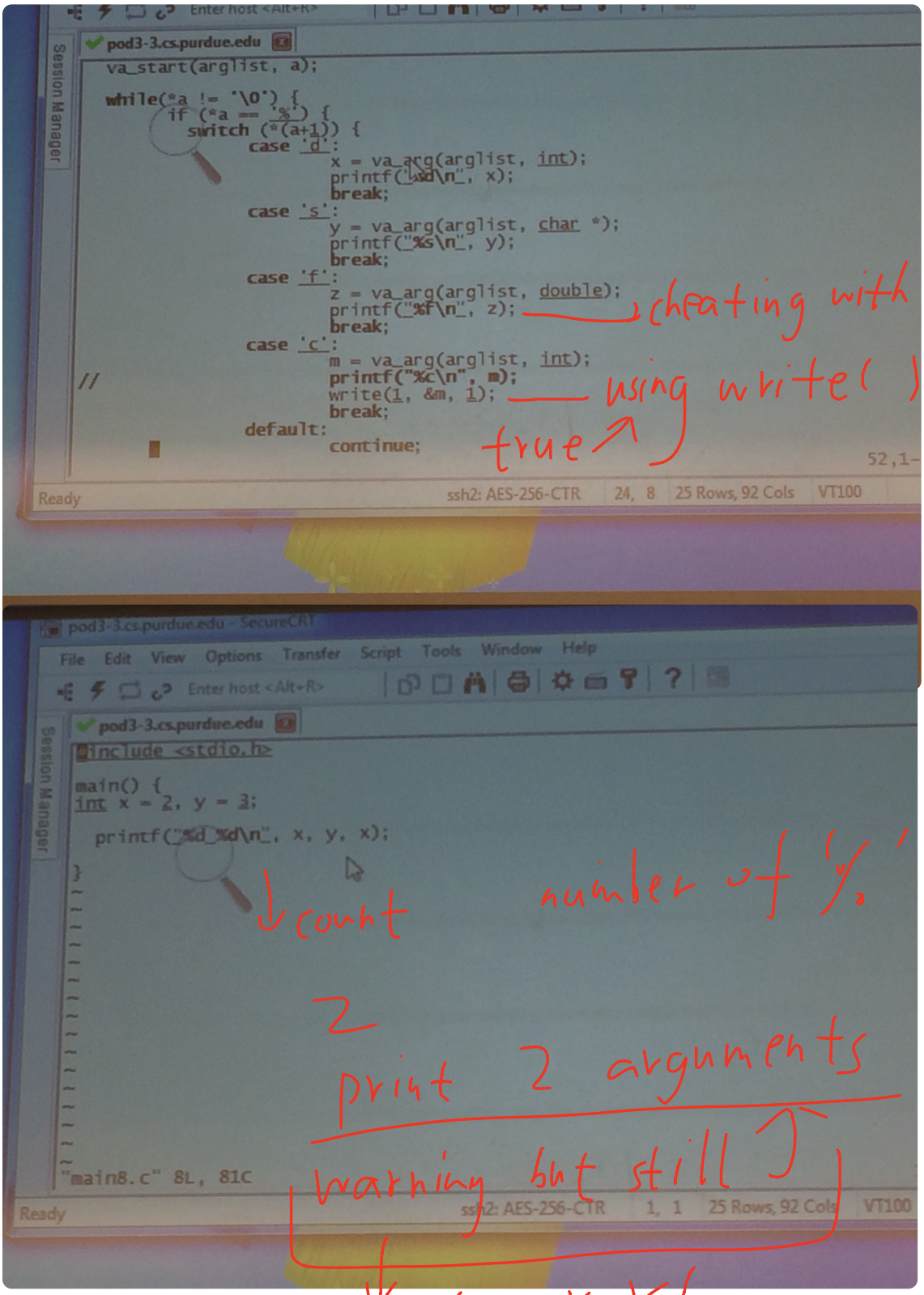

pod3-3.cs.purdue.edu
Session Manager
int res;
res = mysum("iizdzd", 2, 2, 1, 10, 5, 5);
printf("result = %d\n", res);
}
int mysum(char *a, ...)
{
int sum = 0, i;
int cnt;
va_list arglist;
va_start(arglist, a);
cnt = strlen(a);
printf("string length: %d\n", cnt);
for (i=0; i<cnt; i++)
sum += va_arg(arglist, int);
va_end(arglist);
return sum;
Ready
ssh2: AES-256-C

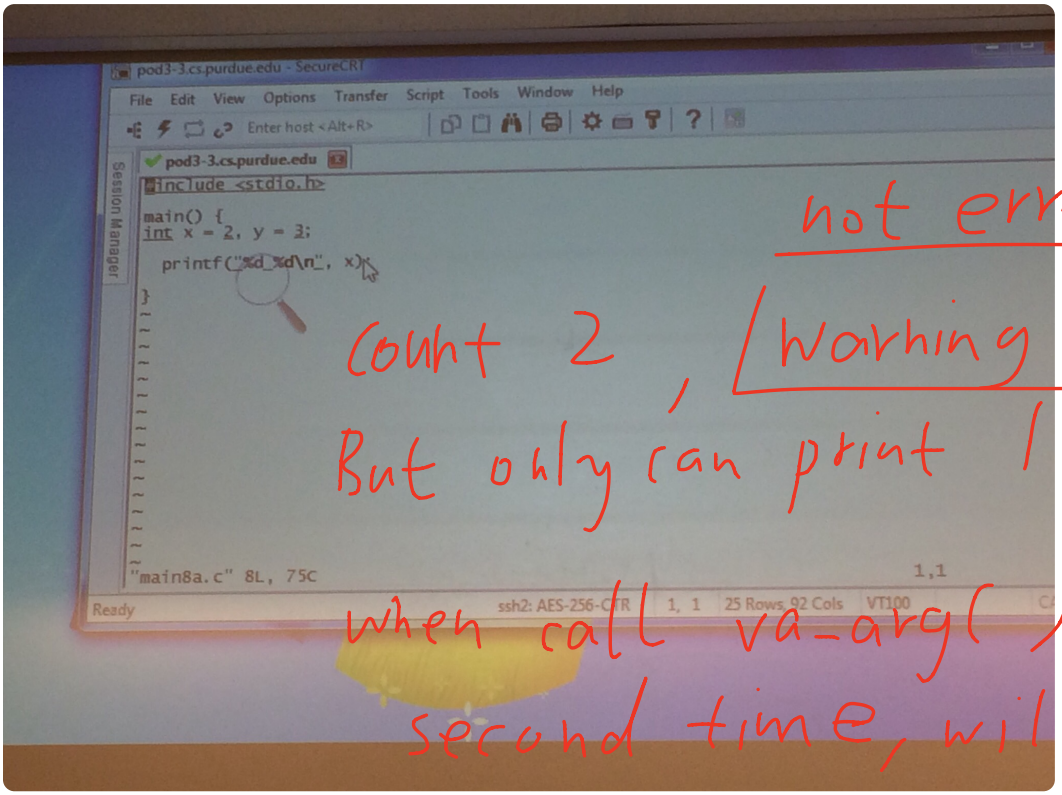

pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window
Enter host <Alt+R>
pod3-3.cs.purdue.edu
Session Manager
{
int mysum(char *, ...);
int res;
res = mysum("abc", 2, "2", 1);
printf("result = %d\n", res);
}
int mysum(char *a, ...)
{
int sum = 0, i;
int cnt;
char *s;
va_list arglist;
va_start(arglist, a);
cnt = strlen(a);
for (i=0; i<cnt; i++) {
if (i == 1) {
s = va_arg(arglist, char *);
sum += atoi(s);
Ready
ssh2: AES
pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window
Enter host <Alt+R>
pod3-3.cs.purdue.edu
Session Manager
}
int mysum(char *a, ...)
{
int sum = 0, i;
int cnt;
char *s;
va_list arglist;
va_start(arglist, a);
cnt = strlen(a);
for (i=0; i<cnt; i++) {
if (i == 1) {
s = va_arg(arglist, char *);
sum += atoi(s);
}
else
sum += va_arg(arglist, int);
}
va_end(arglist);
we know the third argument is a string
Ready
ssh2: AES-256-C

myprintf


pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
pod3-3.cs.purdue.edu
Session Manager
// example to illustrate different number of ar
// a string as in printf() and counts the numb
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <stdlib.h>
main()
{
int myprintf(char *, ...);
myprintf("%d %d %d", 2, 2, 5);
}
int myprintf(char *a, ...)
{
int i, count = 0;
int s;
va_list arglist;
va_start(arglist, a);
"main6.c" 41L, 593C
Ready
ssh2: AES-256-
count
the number
of '%'
pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
pod3-3.cs.purdue.edu
Session Manager
int myprintf(char *a, ...)
{
int i, count = 0;
int s;
va_list arglist;
va_start(arglist, a);
while(*a != '\0') {
if (*a == '%')
count++;
a++;
}
printf("count = %d\n", count);
for (i=0; i<count; i++) {
s = va_arg(arglist, int);
printf("%d\n", s);
}
va_end(arglist);
write( )
Ready
ssh2: AES-256-CTR


pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Session Manager
pod3-3.cs.purdue.edu
// example to illustrate different number of arguments: first argument is
// a string as in printf(), counts the number of occurrences of %, and
// checks for type of input: %d, %s, %d, %c
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <stdlib.h>
main()
{
char r = 'y';
int h = 2;
int myprintf(char *, ...);
myprintf("%c_%d_%s_%d_%f_%c", r, h, "hello", 5, 7.2, 'y');
}
int myprintf(char *a, ...)
{
int x;
char *y;
"main7.c" 60L, 962C
1,1
Enter host <Alt+R>
Session Manager
pod3-3.cs.purdue.edu
int myprintf(char *a, ...)
{
int x;
char *y;
double z;
char m;
va_list arglist;
va_start(arglist, a);
while(*a != '\0') {
if (*a == '%') {
switch (*(a+1)) {
case 'd':
x = va_arg(arglist, int);
printf("%d\n", x);
break;
case 's':
y = va_arg(arglist, char *);
printf("%s\n", y);
break;
case 'f':
z = va_arg(arglist, double);
43,1-8
Ready
ssh2: AES-256-CTR 24, 8 25 Rows, 92 Cols VT100


pod3-3.cs.purdue.edu
va_start(arglist, a);
while(*a != '\0') {
if (*a == '%') {
switch (*(a+1)) {
case 'd':
x = va_arg(arglist, int);
printf("%d\n", x);
break;
case 's':
y = va_arg(arglist, char *);
printf("%s\n", y);
break;
case 'f':
z = va_arg(arglist, double);
printf("%f\n", z);
break;
case 'c':
m = va_arg(arglist, int);
printf("%c\n", m);
write(1, &m, 1);
break;
default:
continue;
//
52,1-
Ready
ssh2: AES-256-CTR
24, 8
25 Rows, 92 Cols
VT100
cheating with printf
using write( )
true
pod3-3.cs.purdue.edu - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
pod3-3.cs.purdue.edu
#include <stdio.h>
main() {
int x = 2, y = 3;
printf("%d %d\n", x, y, x);
}
"main8.c" 8L, 81C
Ready
ssh2: AES-256-CTR
1, 1
25 Rows, 92 Cols
VT100
count
number of '%'
2
print 2 arguments
warning but still
not error!



not error

count 2, [warning!!!]

But only can print 1 correctly

when call va_arg() the second time, will print junk!