RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Recitation 2

Jae Woo Joo

Programming Assignment 1 (Tar/Untar)

- How to tar
  - e.g. $ tar cvf [filename].tar dir1 dir2 file1 file2
  - For your assignment
    - Go to the parent directory of pa1
    - $ tar cvf pa1.tar pa1

- How to untar
  - Go to the directory that has .tar file
  - $ tar xvf pa1.tar

Programming Assignment 1 (Makefile)

- What is Makefile?
  - A special file, containing shell command, that you create and name Makefile
  - If you type “make”, then the commands in the Makefile will be executed

- Makefile contains a list of rules
  - Target: dependencies

    [TAB] Action line (which are Commands)

- Follow the example on sakai

```
all: first
first: first.c
        gcc –Wall –Werror –fsanitize=address first.c –o first
clean:
        rm –rf first
```

Programming Assignment 1 (Auto grader)

```
autograder
        pa1
           | - first
                | -- first.c
                | -- first.h (if used)
                | -- Makefile
           .
           .
           | - ninth
                 | -- ninth.c
                 | -- ninth.h (if used)
                 | -- Makefile
```

Programming Assignment 1 (Auto grader)

- Untar autograder.tar
  - $ tar xvf autograder.tar
- Copy pa1 into autograder directory
  - $ cp –r pa1 autograder
- Go to autograder directory
  - $ cd autograder
- Run autograder
  - $ python auto_grader.py

Programming Assignment 1 (Submission)

```
pa1
    | - first
          | -- first.c
          | -- first.h (if used)
          | -- Makefile
    .
    .
    | - ninth
           | -- ninth.c
           | -- ninth.h (if used)
           | -- Makefile
```

GDB

- Provides extensive facilities for tracing program execution
  - Step through program line at a time
  - Monitor / modify internal variables

- You need **to compile** your code with **–g** flag
  - gcc –g foo.c –o foo

GDB

- Then we use gdb
  - $ gdb [executable program name]
- Debug
  - (gdb) run
- End debugging
  - (gdb) q or quit
- Observe source code
  - (gdb) l or (list) or list 10
  - Could change the number of lines => (gdb) set listsize [num]

GDB

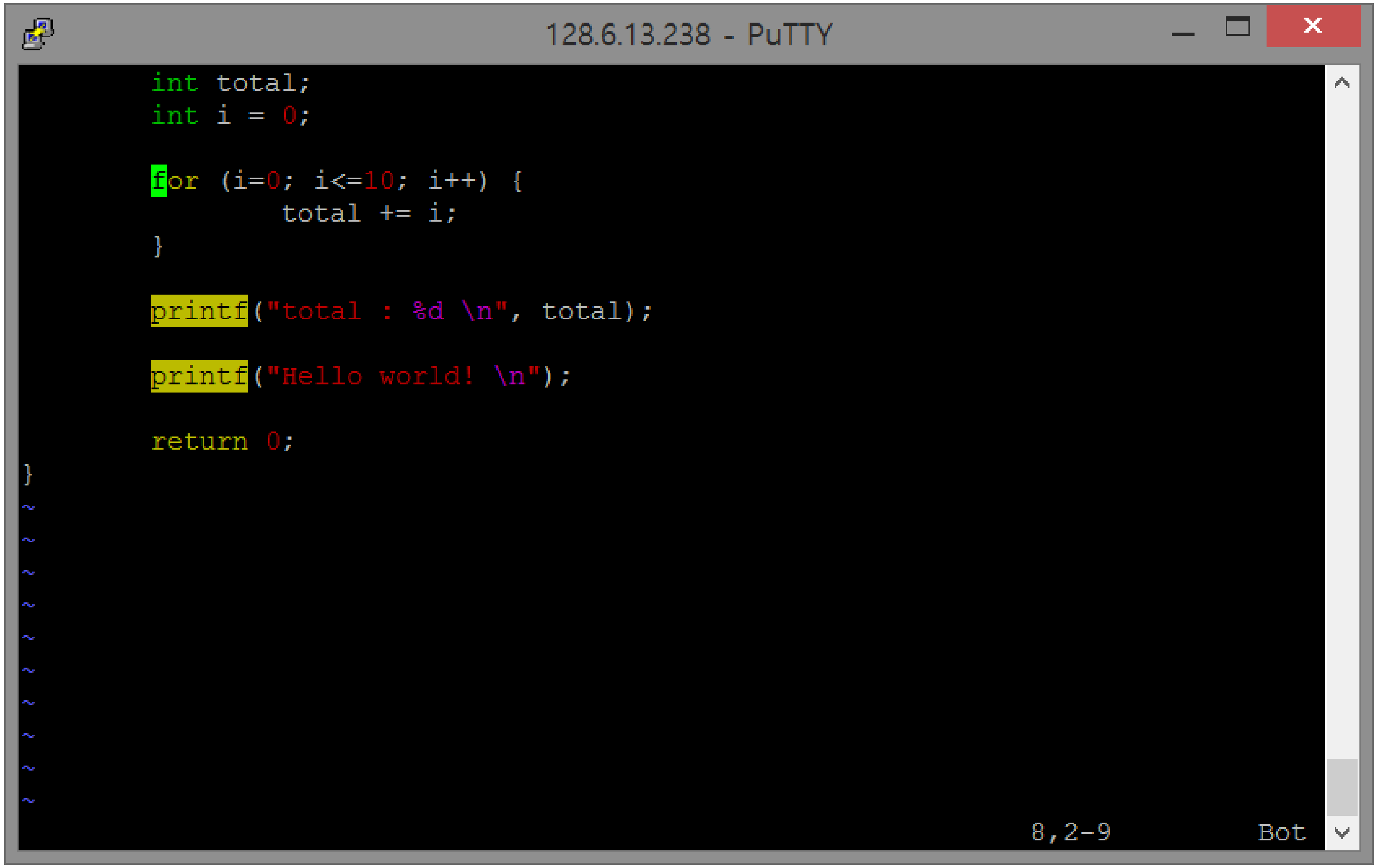
- Setting breakpoints
  - (gdb) break [function name]
  - (gdb) break [line num]

- Clearing breakpoints
  - (gdb) clear [function name]
  - (gdb) clear [line num]
  - (gdb) delete => clearing all breakpoints

GDB

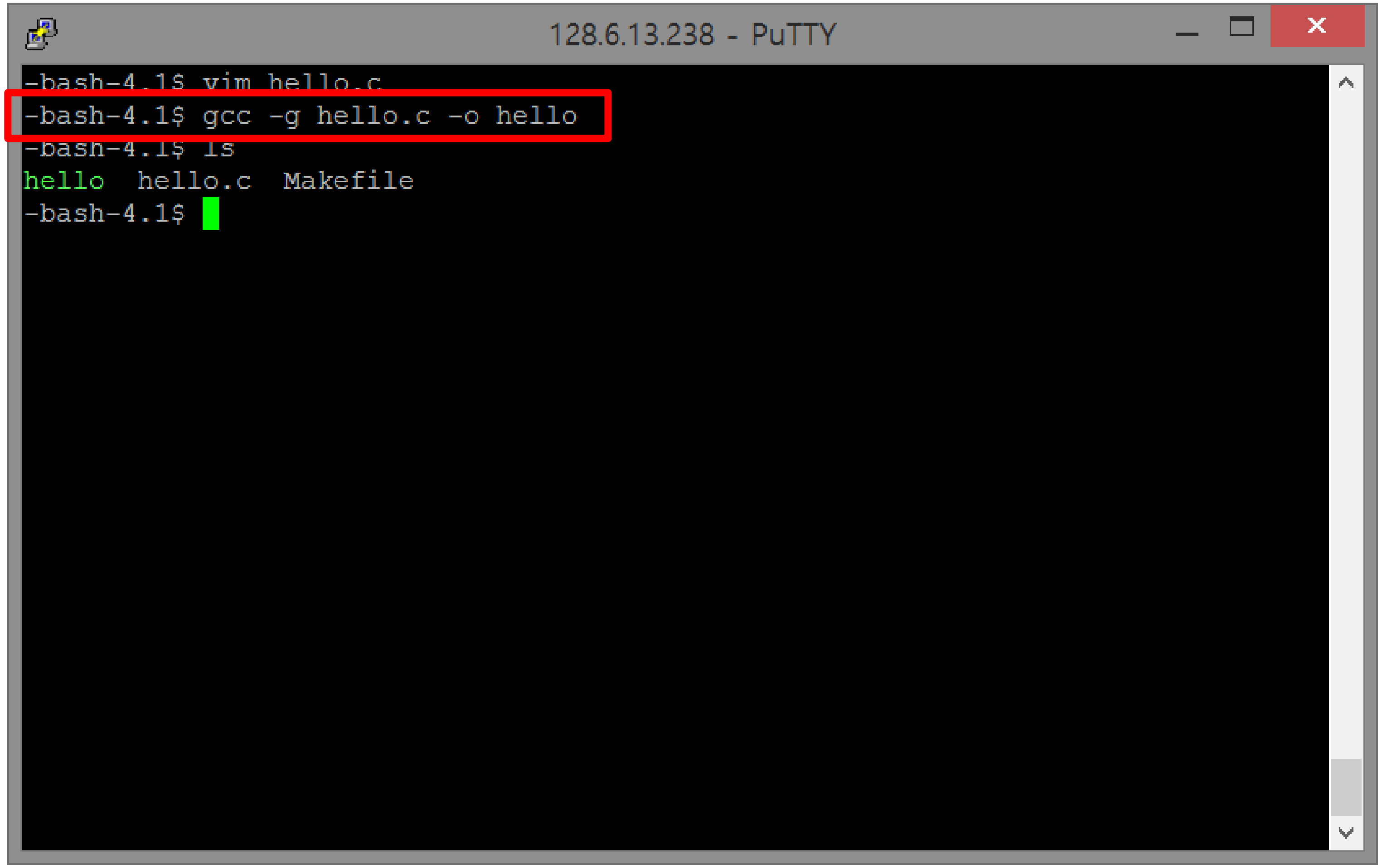
- Printing variables
  - (gdb) print [variable]
  - (gdb) display [variable]
- Going step by step
  - (gdb) next
- More information
  - http://www.yolinux.com/TUTORIALS/GDB-Commands.html#GDB_COMMAND_LINE_ARGS

GDB



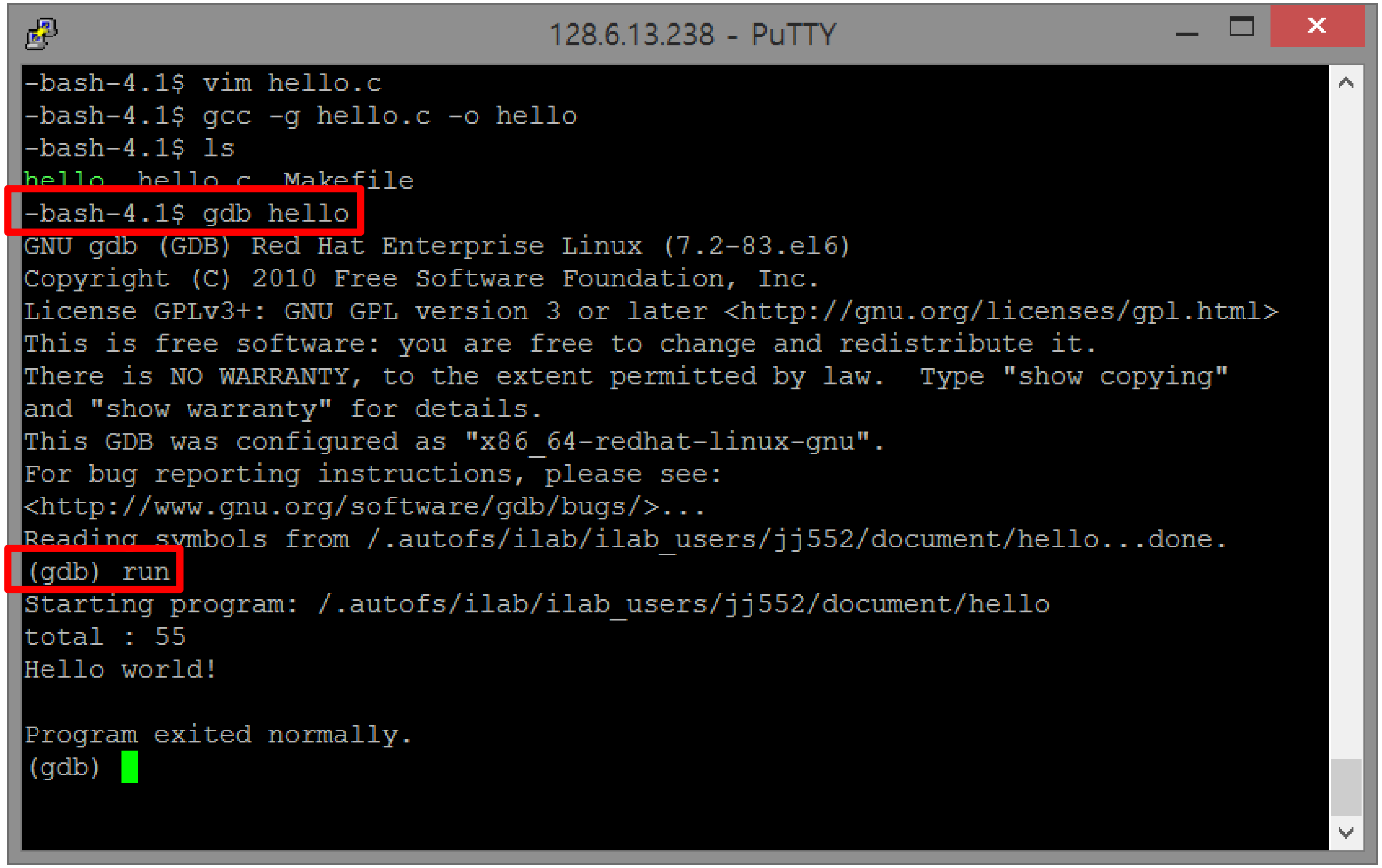
```
        int total;
        int i = 0;

        for (i=0; i<=10; i++) {
                total += i;
        }

        printf("total : %d \n", total);

        printf("Hello world! \n");

        return 0;
}
~
~
~
~
~
~
~
~
~
~
                                                                            8,2-9         Bot
```

GDB


128.6.13.238 - PuTTY

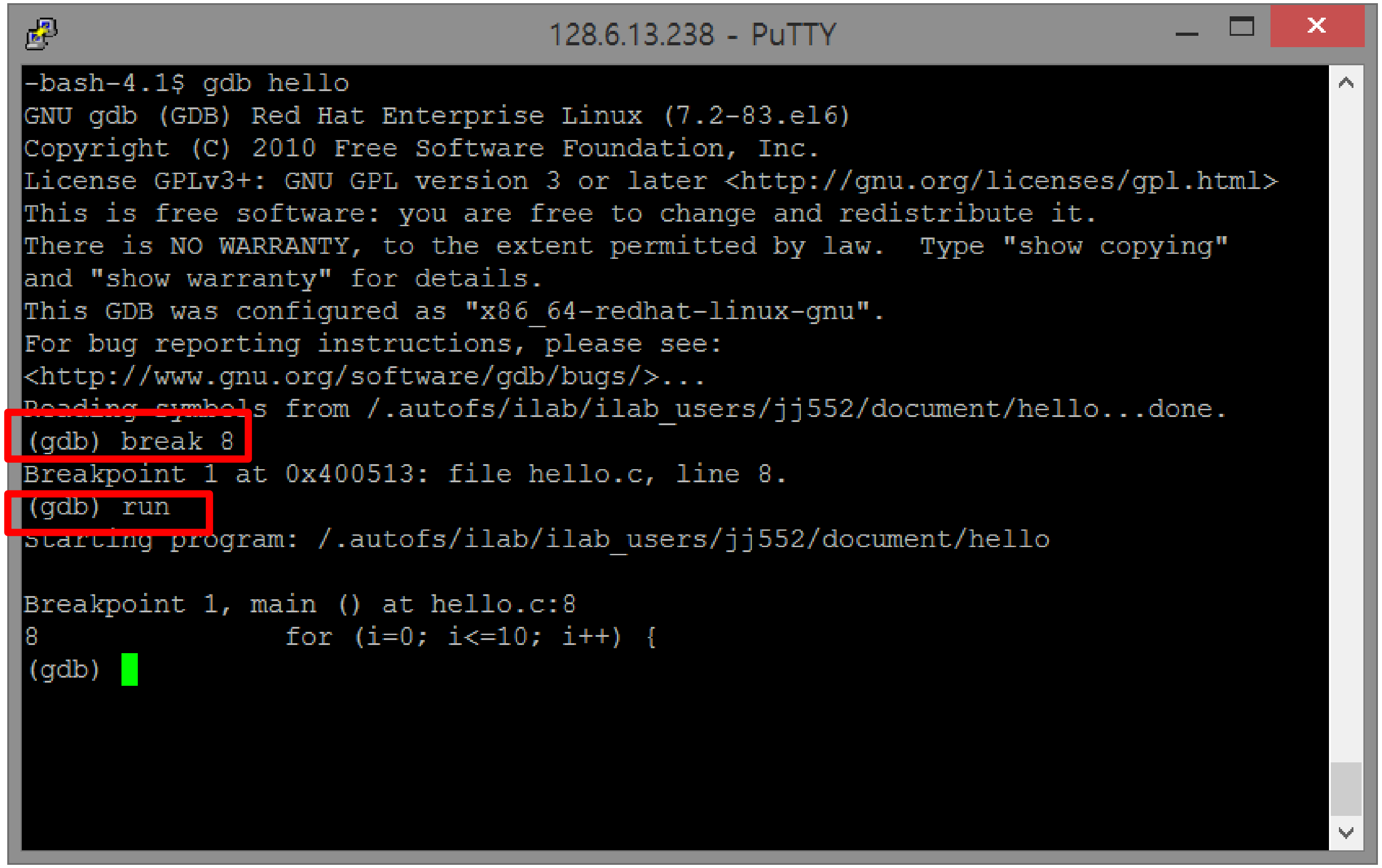
```
-bash-4.1$ vim hello.c
-bash-4.1$ gcc -g hello.c -o hello
-bash-4.1$ ls
hello  hello.c  Makefile
-bash-4.1$
```

GDB



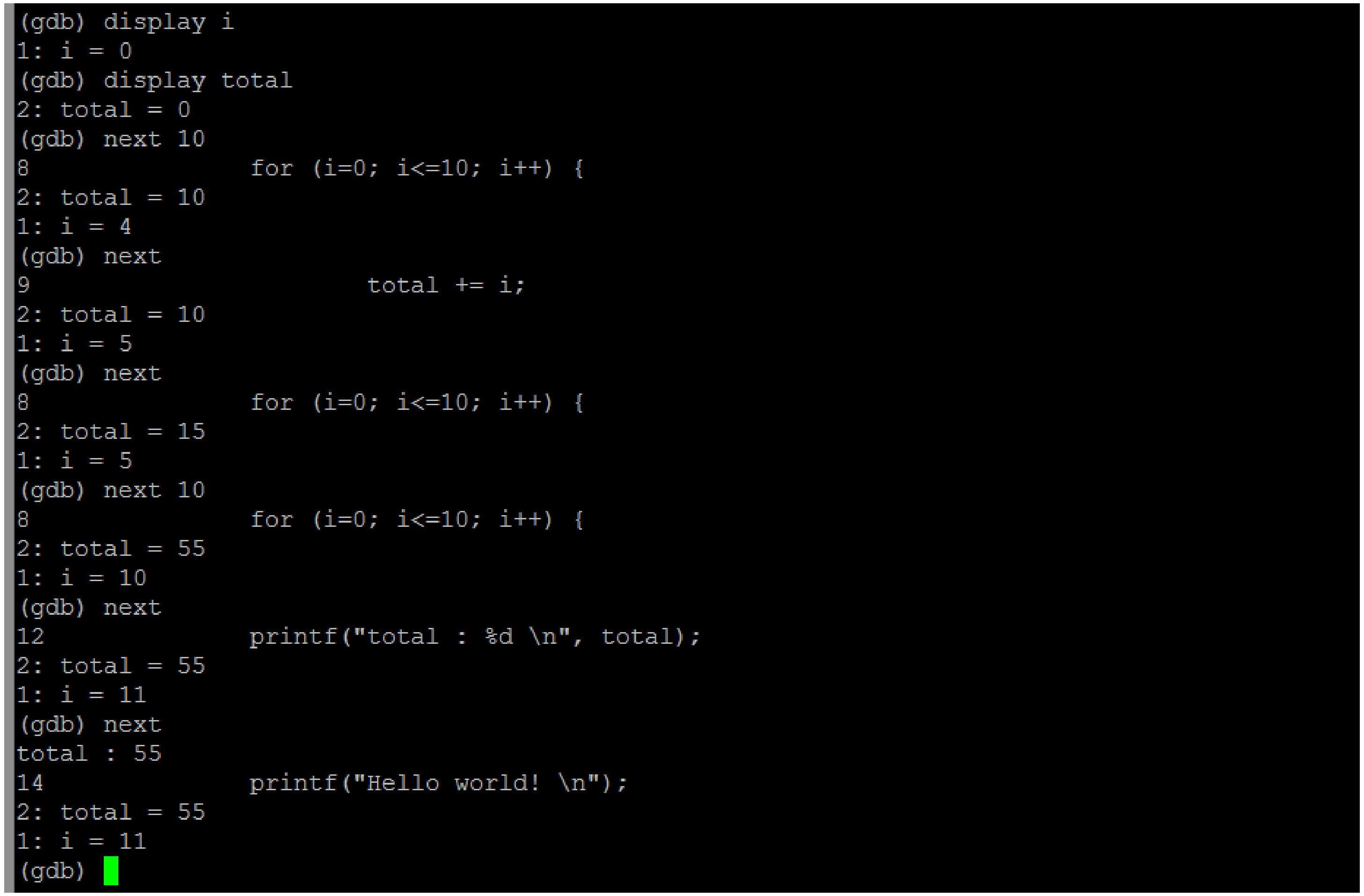
```
-bash-4.1$ vim hello.c
-bash-4.1$ gcc -g hello.c -o hello
-bash-4.1$ ls
hello  hello.c  Makefile
-bash-4.1$ gdb hello
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-83.el6)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /.autofs/ilab/ilab_users/jj552/document/hello...done.
(gdb) run
Starting program: /.autofs/ilab/ilab_users/jj552/document/hello
total : 55
Hello world!

Program exited normally.
(gdb)
```

GDB



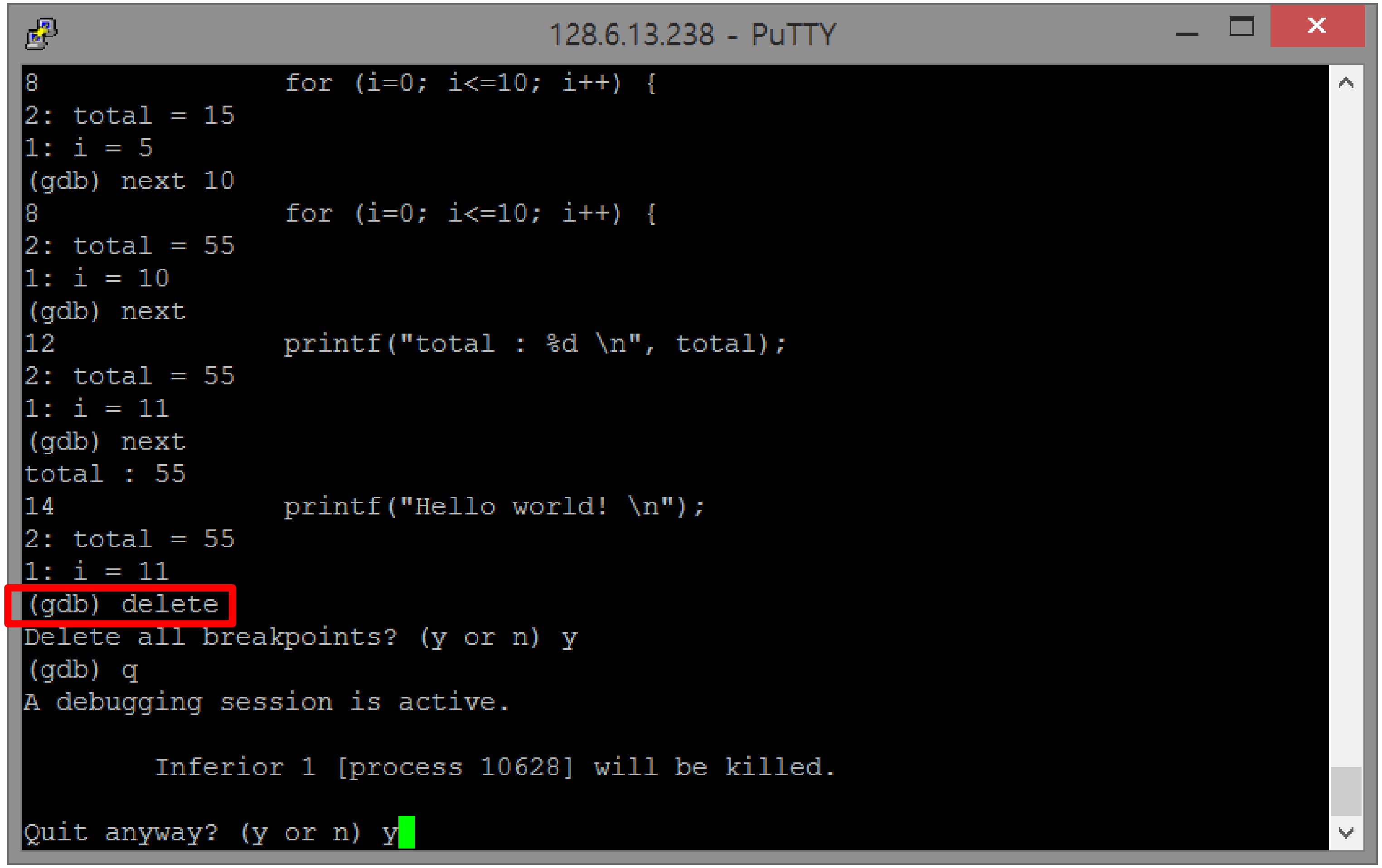
```
-bash-4.1$ gdb hello
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-83.el6)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /.autofs/ilab/ilab_users/jj552/document/hello...done.
(gdb) break 8
Breakpoint 1 at 0x400513: file hello.c, line 8.
(gdb) run
Starting program: /.autofs/ilab/ilab_users/jj552/document/hello

Breakpoint 1, main () at hello.c:8
8               for (i=0; i<=10; i++) {
(gdb)
```

GDB



```
(gdb) display i
1: i = 0
(gdb) display total
2: total = 0
(gdb) next 10
8               for (i=0; i<=10; i++) {
2: total = 10
1: i = 4
(gdb) next
9                       total += i;
2: total = 10
1: i = 5
(gdb) next
8               for (i=0; i<=10; i++) {
2: total = 15
1: i = 5
(gdb) next 10
8               for (i=0; i<=10; i++) {
2: total = 55
1: i = 10
(gdb) next
12              printf("total : %d \n", total);
2: total = 55
1: i = 11
(gdb) next
total : 55
14              printf("Hello world! \n");
2: total = 55
1: i = 11
(gdb)
```

GDB



```
8               for (i=0; i<=10; i++) {
2: total = 15
1: i = 5
(gdb) next 10
8               for (i=0; i<=10; i++) {
2: total = 55
1: i = 10
(gdb) next
12              printf("total : %d \n", total);
2: total = 55
1: i = 11
(gdb) next
total : 55
14              printf("Hello world! \n");
2: total = 55
1: i = 11
(gdb) delete
Delete all breakpoints? (y or n) y
(gdb) q
A debugging session is active.

        Inferior 1 [process 10628] will be killed.

Quit anyway? (y or n) y
```

Pointer

- Pointer is a variable that can store an address
- **int *numAddr;**
- **numAddr = &num;**

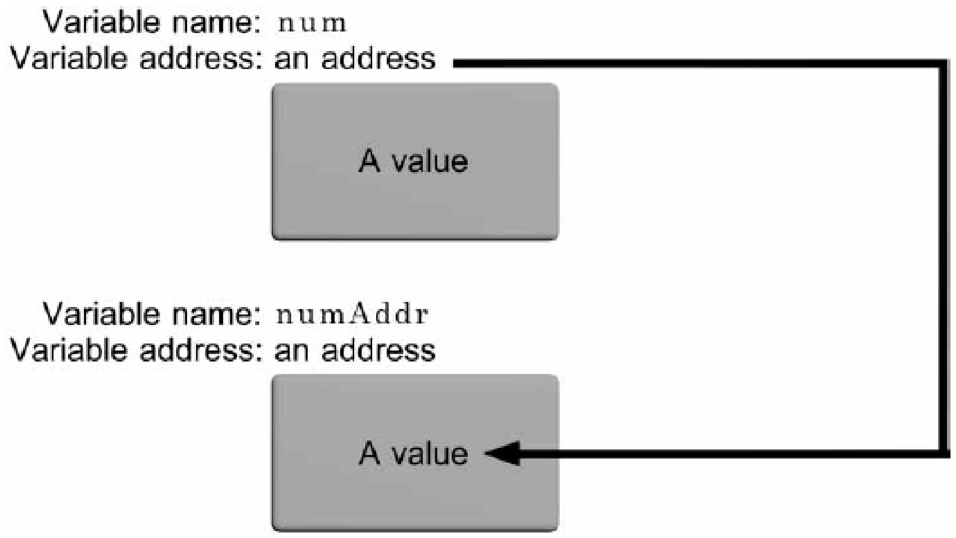


Figure 7.9 Storing num's address in numAddr

Pointer

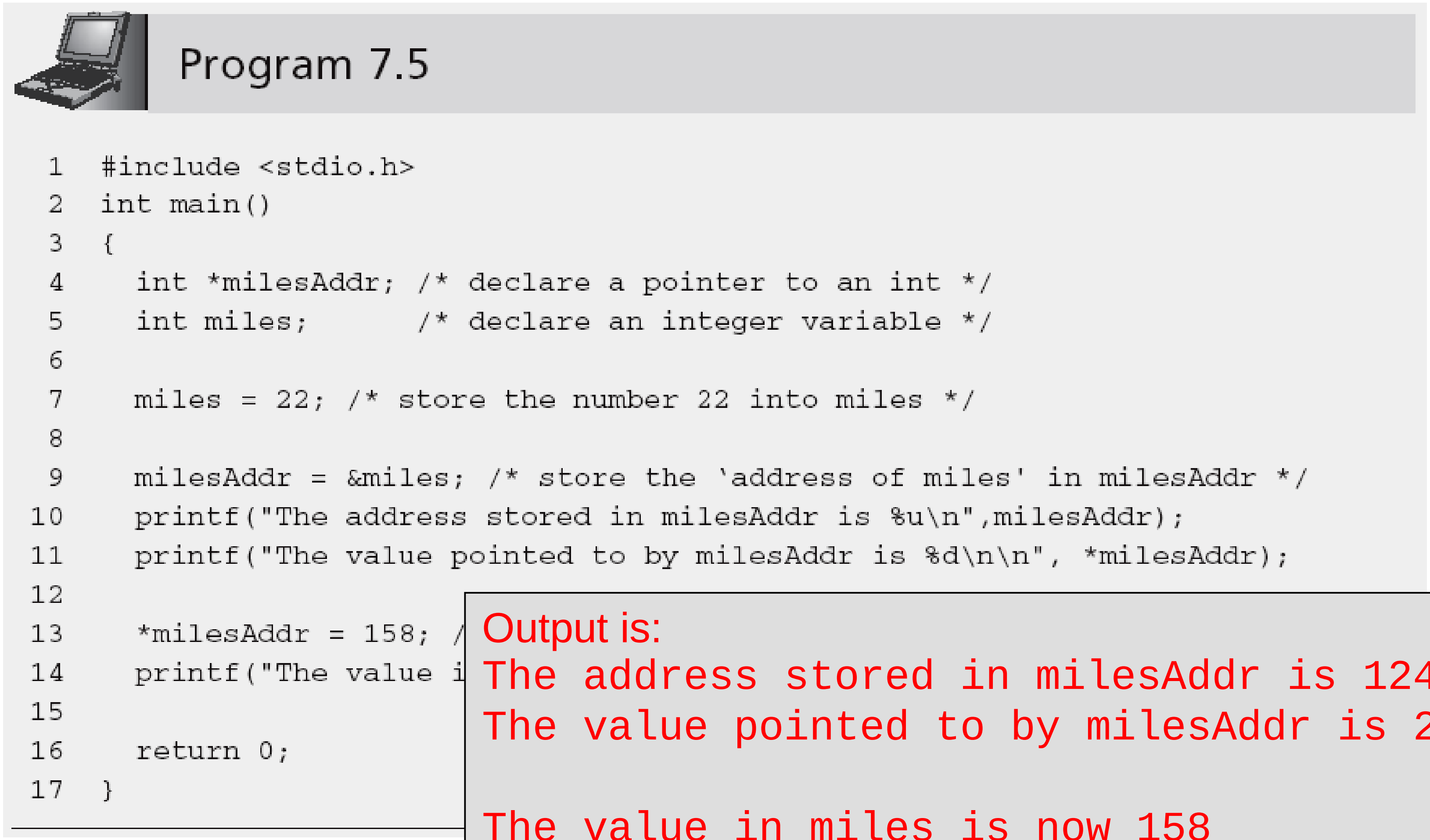
- De-reference: *
  - ***numAddr** means the variable whose address is stored in **numAddr**
  - Or, the variable pointed to by **numAddr**
- Declaring a pointer variable
  - int *numAddr

Pointer

Program 7.5

```
#include <stdio.h>
int main()
{
  int *milesAddr; /* declare a pointer to an int */
  int miles;      /* declare an integer variable */

  miles = 22; /* store the number 22 into miles */

  milesAddr = &miles; /* store the 'address of miles' in milesAddr */
  printf("The address stored in milesAddr is %u\n",milesAddr);
  printf("The value pointed to by milesAddr is %d\n\n", *milesAddr);

  *milesAddr = 158; /* set the value pointed to by milesAddr to 158 */
  printf("The value in miles is now %d\n", miles);

  return 0;
}
```

Pointer



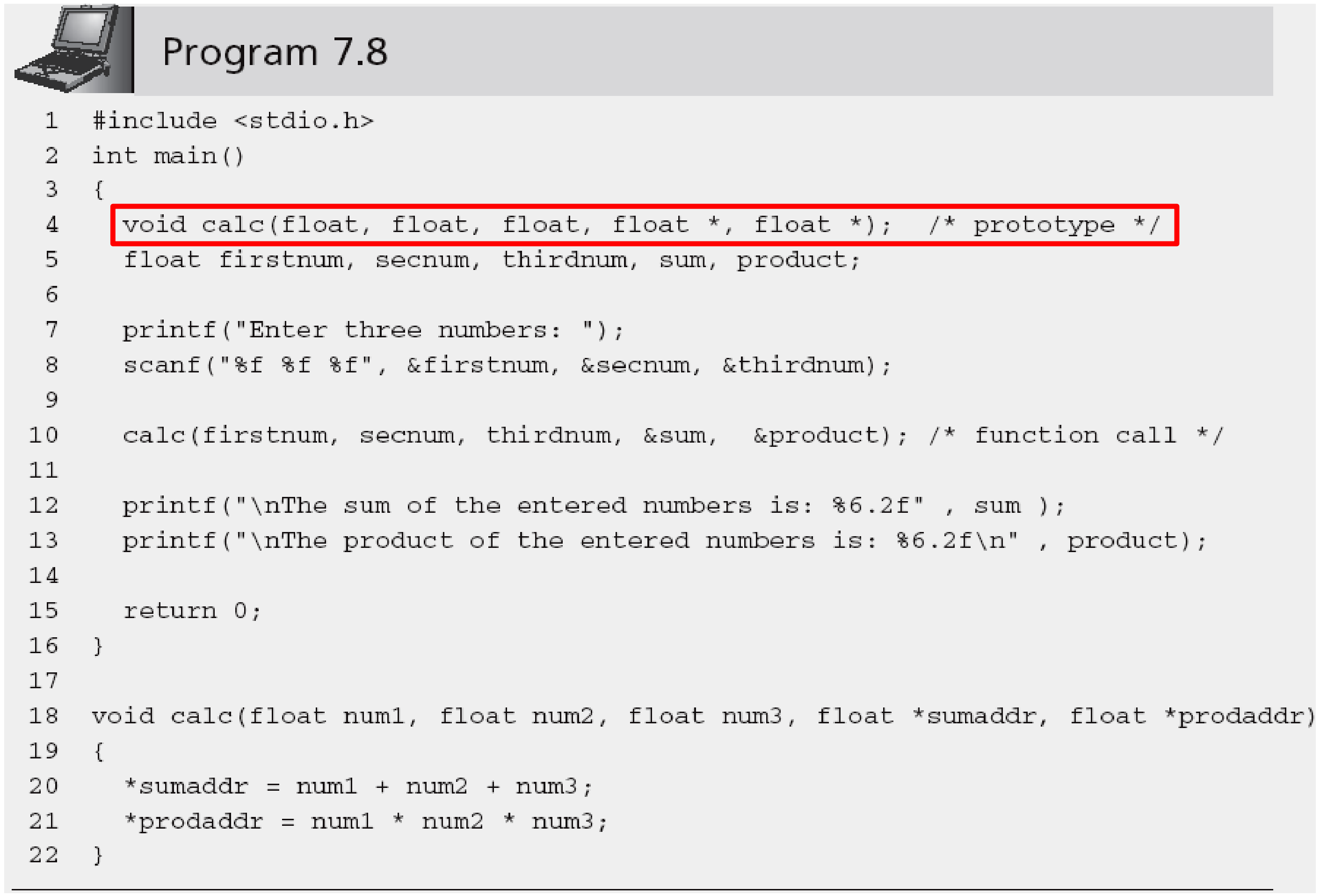
Program 7.5

```
#include <stdio.h>
int main()
{
  int *milesAddr; /* declare a pointer to an int */
  int miles;      /* declare an integer variable */

  miles = 22; /* store the number 22 into miles */

  milesAddr = &miles; /* store the 'address of miles' in milesAddr */
  printf("The address stored in milesAddr is %u\n",milesAddr);
  printf("The value pointed to by milesAddr is %d\n\n", *milesAddr);

  *milesAddr = 158; /
  printf("The value i

  return 0;
}
```

Output is:

```
The address stored in milesAddr is 1244872
The value pointed to by milesAddr is 22

The value in miles is now 158
```

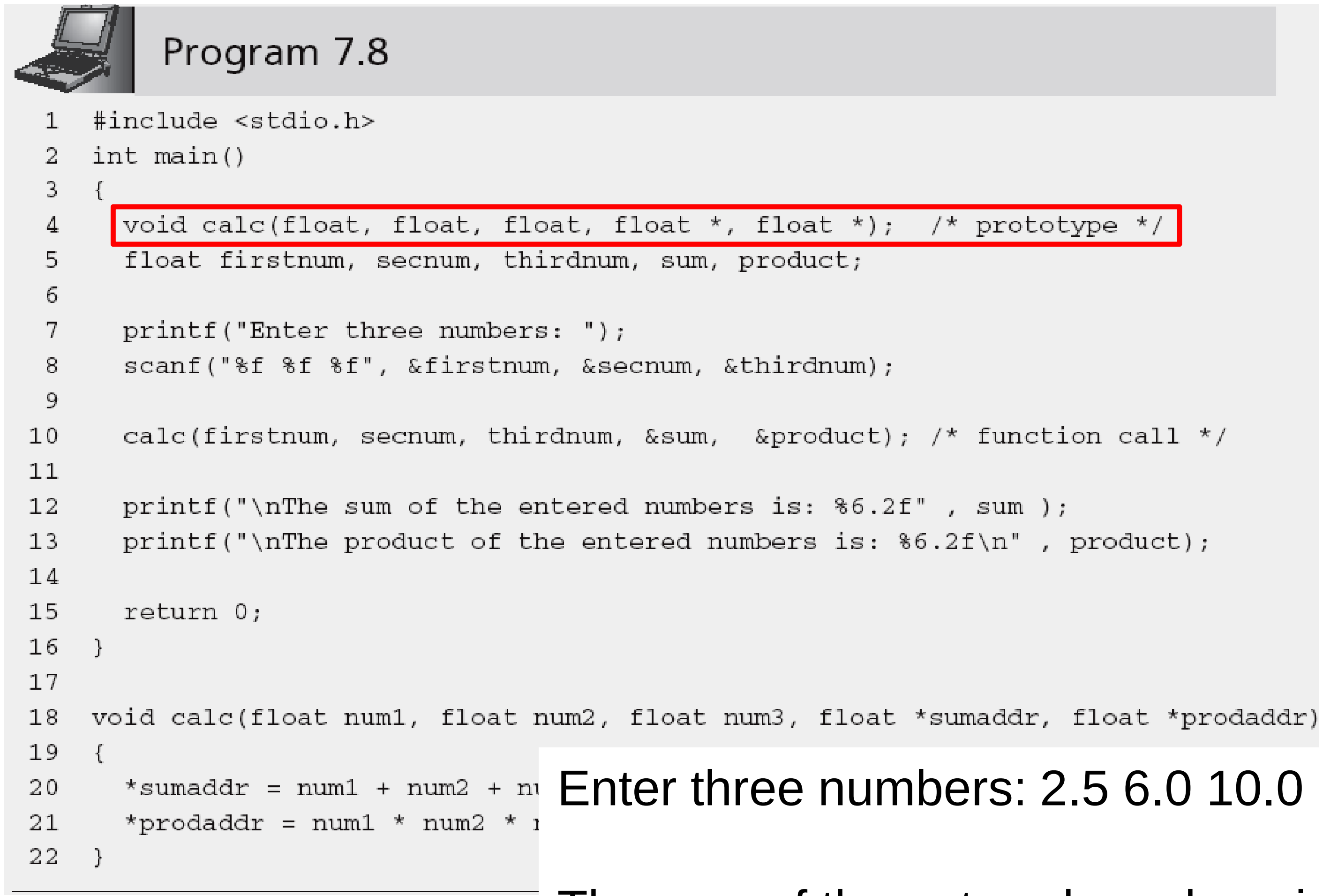
Pointer



Program 7.8

```
#include <stdio.h>
int main()
{
  void calc(float, float, float, float *, float *);  /* prototype */
  float firstnum, secnum, thirdnum, sum, product;

  printf("Enter three numbers: ");
  scanf("%f %f %f", &firstnum, &secnum, &thirdnum);

  calc(firstnum, secnum, thirdnum, &sum,  &product); /* function call */

  printf("\nThe sum of the entered numbers is: %6.2f" , sum );
  printf("\nThe product of the entered numbers is: %6.2f\n" , product);

  return 0;
}

void calc(float num1, float num2, float num3, float *sumaddr, float *prodaddr)
{
  *sumaddr = num1 + num2 + num3;
  *prodaddr = num1 * num2 * num3;
}
```

Pointer



Program 7.8

```
#include <stdio.h>
int main()
{
  void calc(float, float, float, float *, float *);  /* prototype */
  float firstnum, secnum, thirdnum, sum, product;

  printf("Enter three numbers: ");
  scanf("%f %f %f", &firstnum, &secnum, &thirdnum);

  calc(firstnum, secnum, thirdnum, &sum,  &product); /* function call */

  printf("\nThe sum of the entered numbers is: %6.2f" , sum );
  printf("\nThe product of the entered numbers is: %6.2f\n" , product);

  return 0;
}

void calc(float num1, float num2, float num3, float *sumaddr, float *prodaddr)
{
  *sumaddr = num1 + num2 + nu
  *prodaddr = num1 * num2 * n
}
```

```
Enter three numbers: 2.5 6.0 10.0

The sum of the entered numbers is: 18.50
The product of the entered numbers is: 150.
00
```

Pointer

```
void swap (float *numAddr1, float *numAddr2) {
        float temp;

        temp = *numAddr1;
        *numAddr1 = *numAddr2;
        *numAddr2 = temp;
}
```

Array and Pointer

- int grade[] = {98, 87, 92,79, 85};

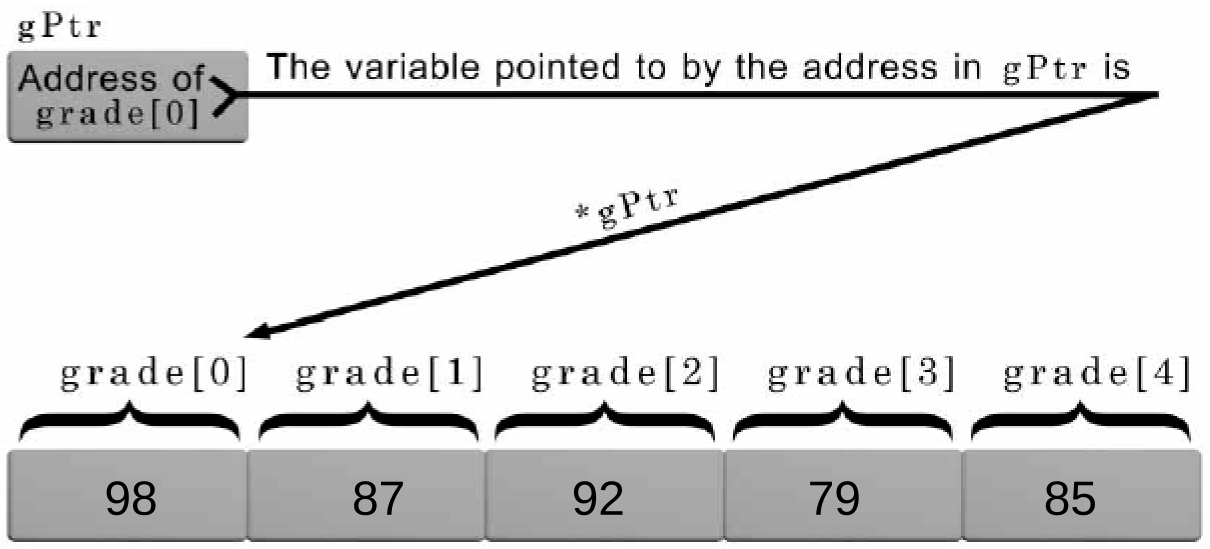


Figure 11.3 The variable pointed to by *gPtr is grade[0]

Array and Pointer

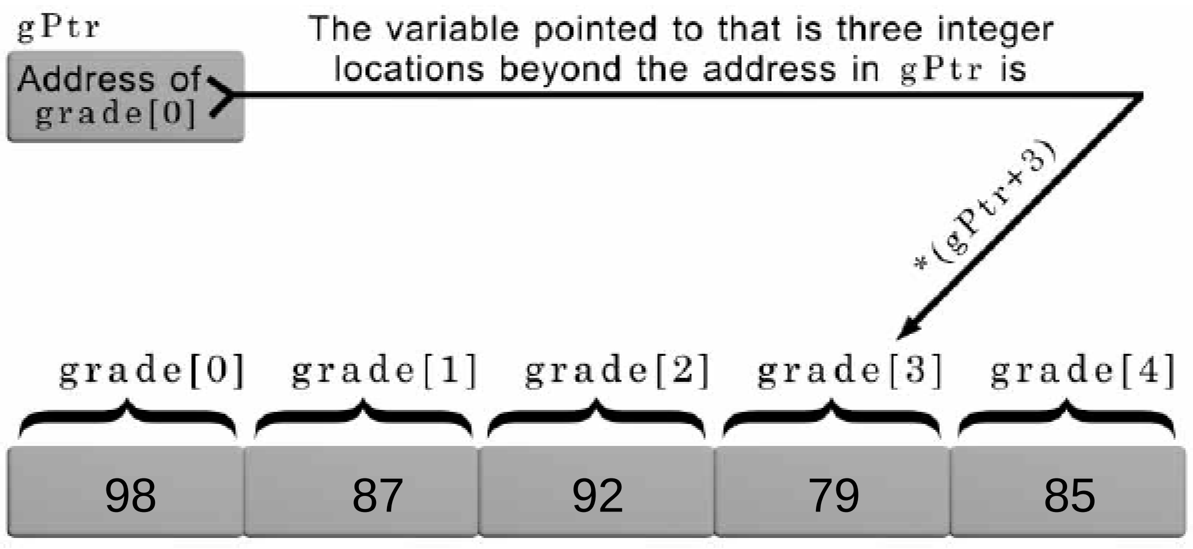

gPtr
Address of grade[0]
The variable pointed to that is three integer locations beyond the address in gPtr is
*(gPtr+3)
grade[0]
grade[1]
grade[2]
grade[3]
grade[4]
98
87
92
79
85

Figure 11.4 An offset of three from the address in gPtr

Array and Pointer

- grade[0] = *gPtr
- grade[1] = *(gPtr + 1)
- grade[2] = *(gPtr + 2)
- grade[3] = *(gPtr + 3)
- grade[4] = *(gPtr + 4)

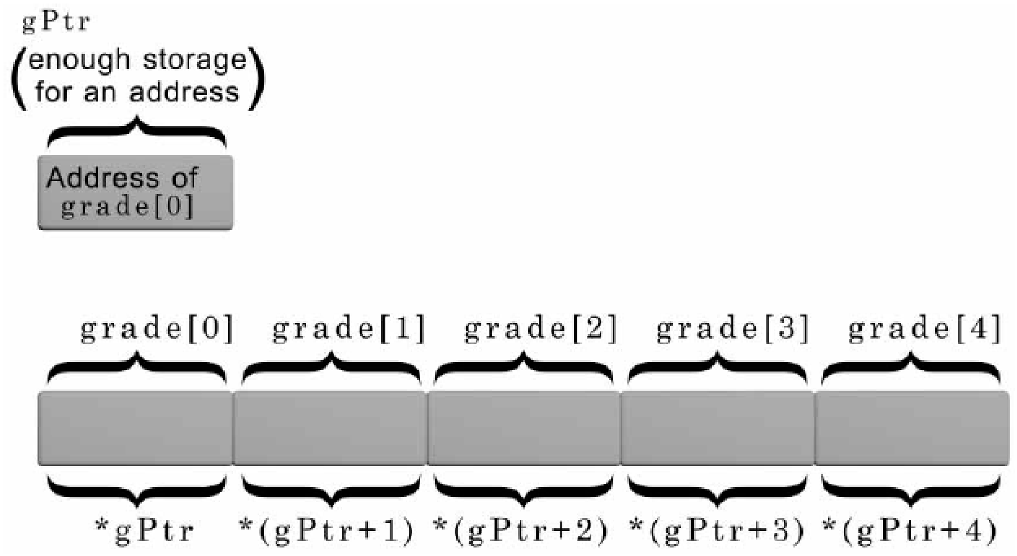


Figure 11.5 The relationship between array elements and pointers

Pointer Arithmetic

```
int nums[100];
int *nPtr;

nPtr = &nums[0];
nPtr = nums;
```

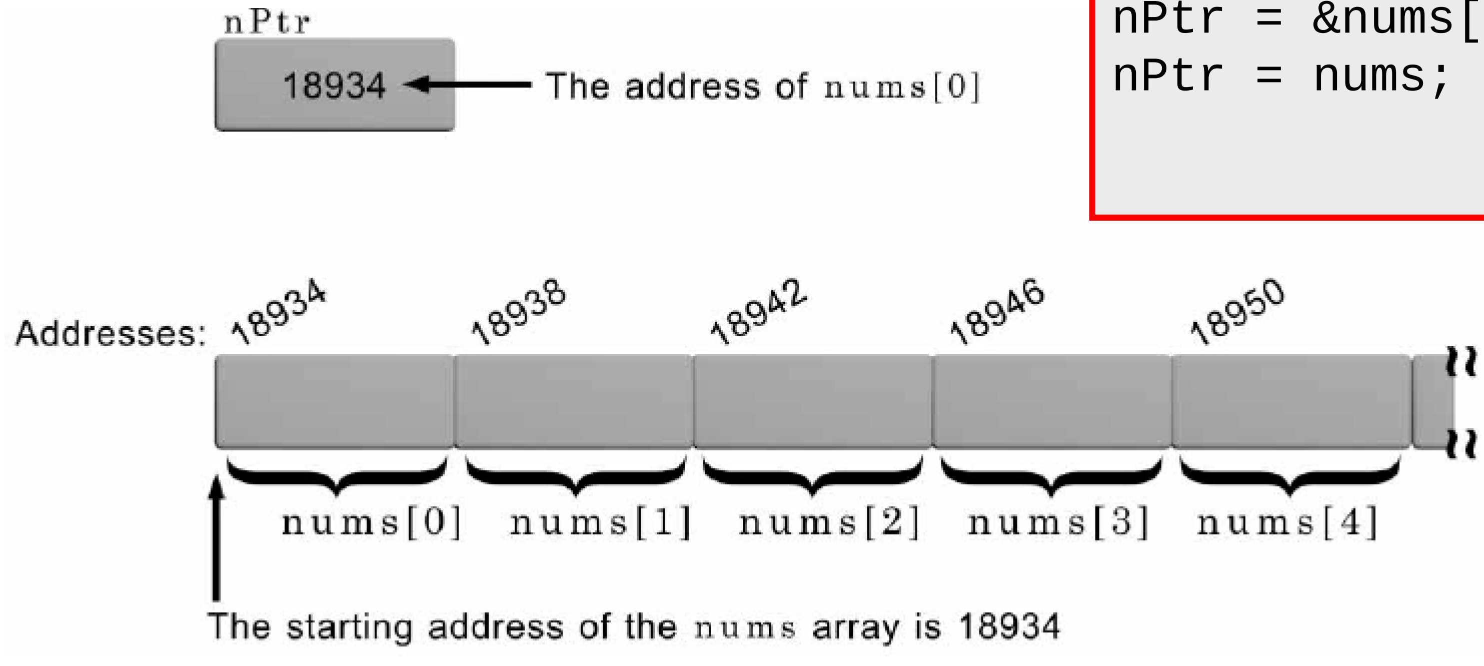


Figure 11.7 The nums array in memory

Pointer Arithmetic

- If **nPtr = &nums[0]**
- nPtr + 2 = &nums[2]
  - nPtr + 2 is the address of nums[2]

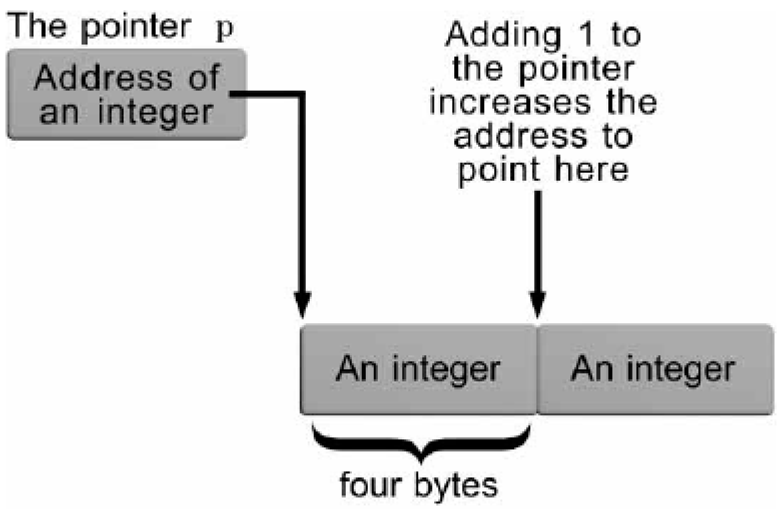


Figure 11.8 Increments are scaled when used with pointers

Pointer arithmetic

- *ptNum++
  - Use the pointer and then increment it
- *++ptNum
  - Increment the pointer before using it
- *ptNum--
  - Use the pointer and then decrement it
- *--ptNum
  - Decrement the pointer before using it

Q&A

- Any questions?