Programming in C

C code tool chain

- Pre-processor (gcc -E)
- Compiler (gcc -c)
- Linker (ld, but invoked via gcc for us)
- Loader (ld-linux.so)

Example

- Open a text editor (geany) and type in the following program:

```
#include "stdio.h"

int main (void)
{
  printf ("Hello World");
  return 0;
}
```

- Save it as (helloworld.c) in a folder called (HelloWorld) on the desktop

Example

- Now, open a terminal and navigate to the folder HelloWorld that you just created

- Type: gcc –c helloworld.c
- List the file contents. What file has been created?

- Type: gcc main.o
- List the file contents. What file has been created?

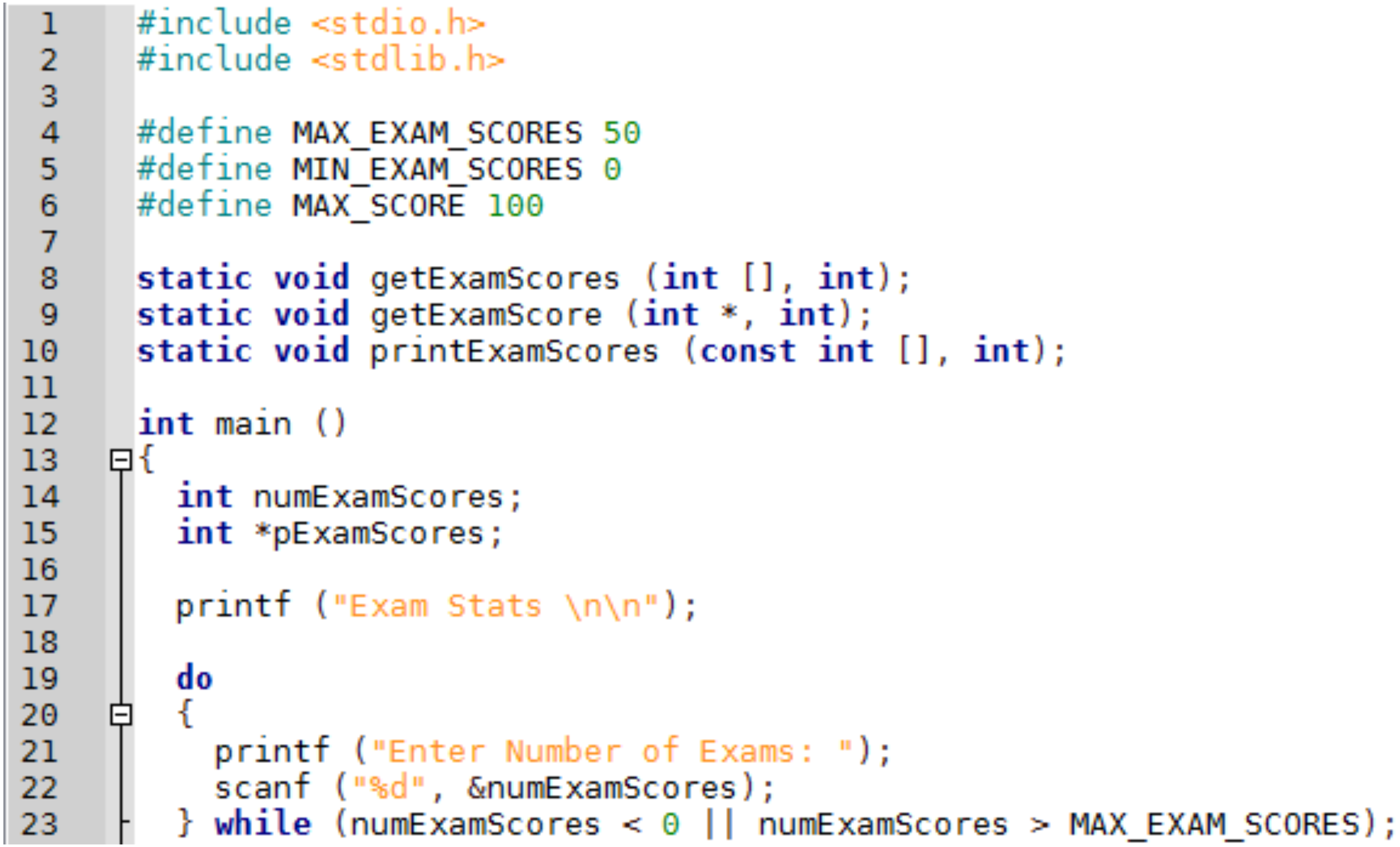
- Type: ./a.out
- What happens?

Define

- Pointer

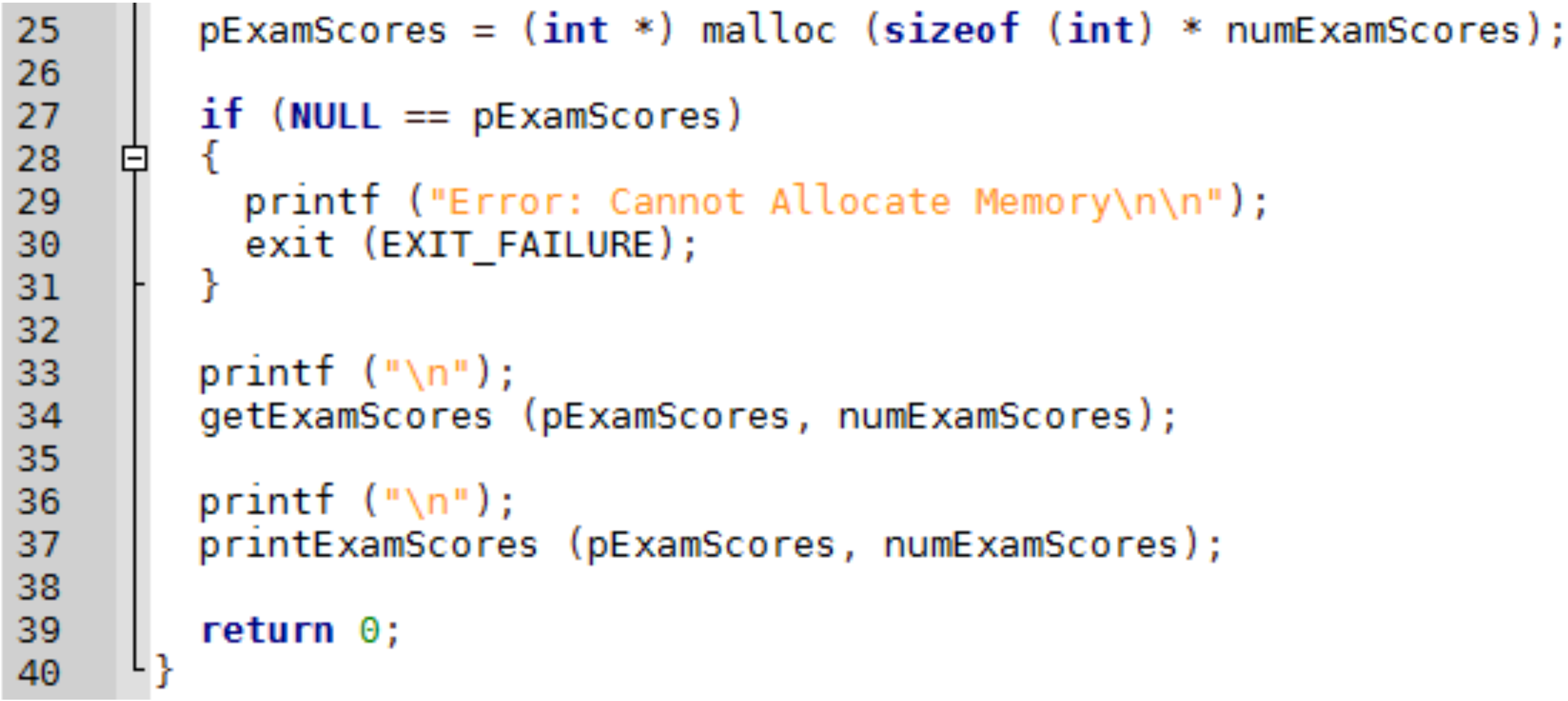
- Memory Address

Loops, Arrays, Pointer Review



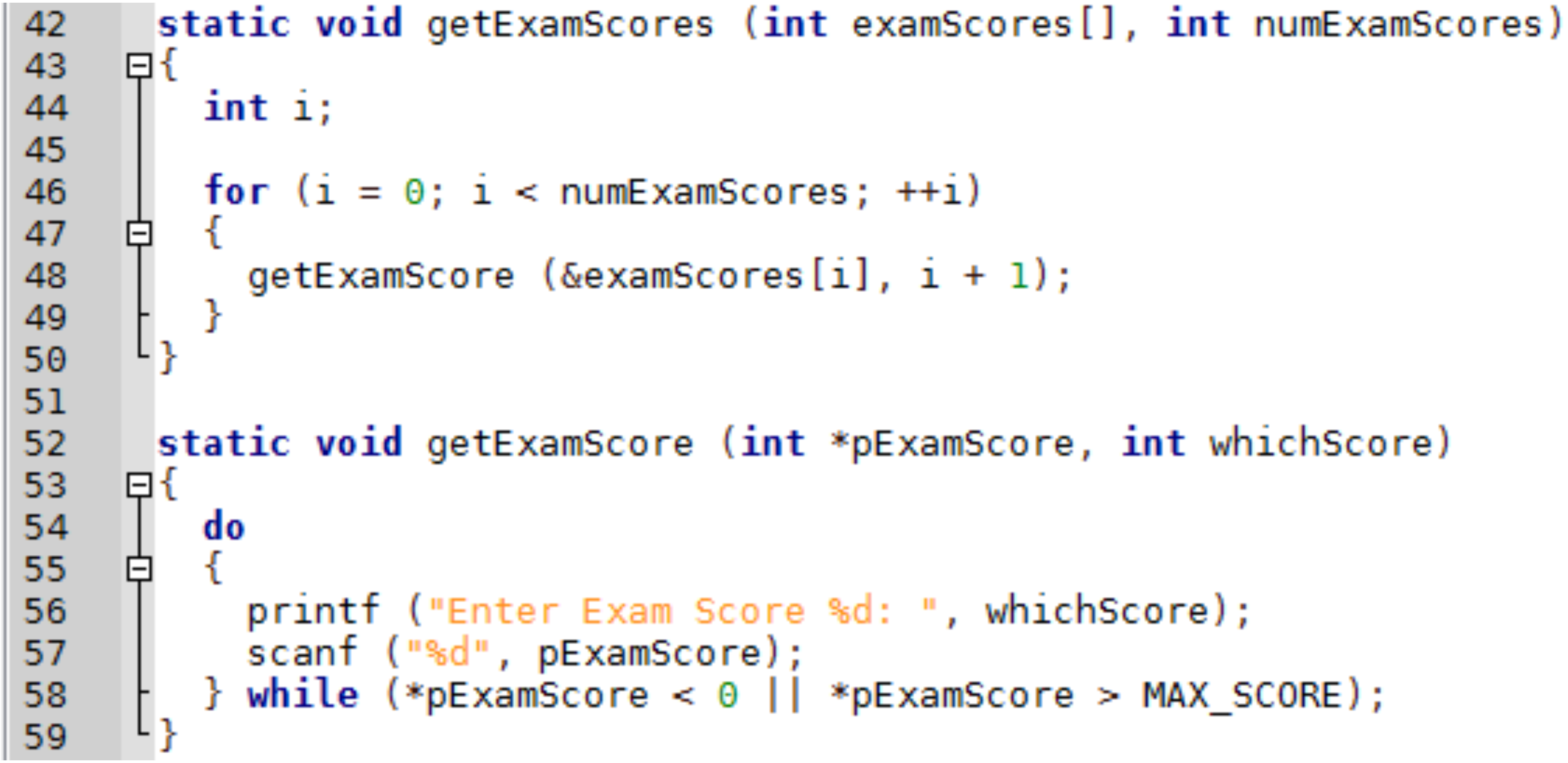
```c
#include <stdio.h>
#include <stdlib.h>

#define MAX_EXAM_SCORES 50
#define MIN_EXAM_SCORES 0
#define MAX_SCORE 100

static void getExamScores (int [], int);
static void getExamScore (int *, int);
static void printExamScores (const int [], int);

int main ()
{
  int numExamScores;
  int *pExamScores;

  printf ("Exam Stats \n\n");

  do
  {
    printf ("Enter Number of Exams: ");
    scanf ("%d", &numExamScores);
  } while (numExamScores < 0 || numExamScores > MAX_EXAM_SCORES);
```

Loops, Arrays, Pointer Review



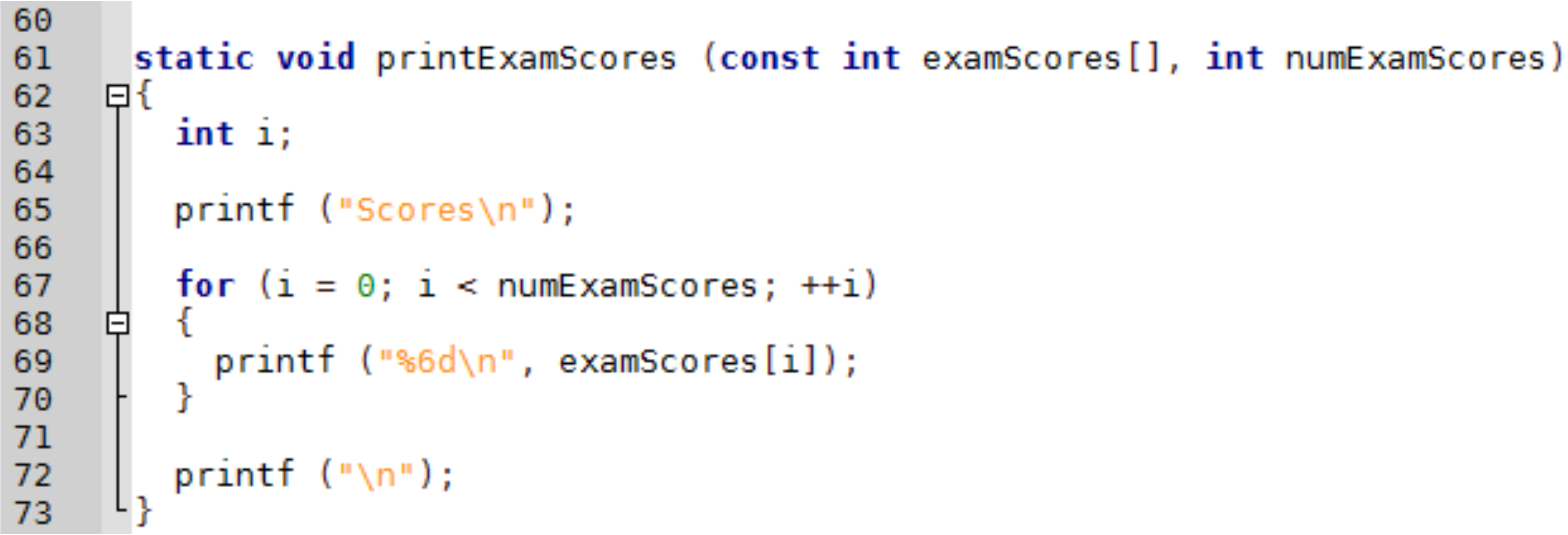
```c
  pExamScores = (int *) malloc (sizeof (int) * numExamScores);

  if (NULL == pExamScores)
  {
    printf ("Error: Cannot Allocate Memory\n\n");
    exit (EXIT_FAILURE);
  }

  printf ("\n");
  getExamScores (pExamScores, numExamScores);

  printf ("\n");
  printExamScores (pExamScores, numExamScores);

  return 0;
}
```

Loops, Arrays, Pointer Review



```c
static void getExamScores (int examScores[], int numExamScores)
{
  int i;

  for (i = 0; i < numExamScores; ++i)
  {
    getExamScore (&examScores[i], i + 1);
  }
}

static void getExamScore (int *pExamScore, int whichScore)
{
  do
  {
    printf ("Enter Exam Score %d: ", whichScore);
    scanf ("%d", pExamScore);
  } while (*pExamScore < 0 || *pExamScore > MAX_SCORE);
}
```

Loops, Arrays, Pointer Review



```c
static void printExamScores (const int examScores[], int numExamScores)
{
  int i;

  printf ("Scores\n");

  for (i = 0; i < numExamScores; ++i)
  {
    printf ("%6d\n", examScores[i]);
  }

  printf ("\n");
}
```

Questions on the Previous Program

- What is the purpose of the **`#define`** statement and who does it create work for:
  - The pre-processor
  - The compiler
  - The linker
  - The loader
- What is the purpose of using **`static`** in a function prototype?
- Explain scanf
- What does "%6d" mean?

Homework

- Type up the program on slides 6-9, then compile, link, and run it
- Turn in a print out answering the following:
  - Does the program run as expected?
  - What input did you use?
  - What output did you get?
  - What questions do you have about the program code?