MIPS TO C-CODE CONVERSION

Given some MIPS assembly, this program will attempt to convert it to C. The goal of this project is to support de-compilation projects, which aim to write C code that yields byte-identical output when compiled with a particular build system. It primarily focuses on supporting popular compilers of the late 1990's.However, it may also work with other compilers or hand-written assembly. The focus of `mips-to-c` is to aid in the process of producing "matching" C source files. These differentiates it from other de-compilation suites, such as IDA or Ghidra . Right now the de-compiler is fairly functional, though it sometimes generates suboptimal code (especially for loops).The input is expected to match a particular assembly format, such as that produced by tools like [`mipsdisasm`](https://github.com/queueRAM/sm64tools).

See the `tests/` directory for some example input and output.

Install

This project requires Python 3.6 or later. To install the Python dependencies:

```bash

python3 -m pip install --upgrade pycparser

Optional: If you are on python3.6, you will also need to install "dataclasses"

python3.6 -m pip install --upgrade dataclasses

You might need to install `pip` first; on Ubuntu this can be done with:

```bash

sudo apt update

sudo apt install python3-pip

```

Usage

```bash

python3 mips\_to\_c.py [options] [--context <context file>] [-f <function name>] <asmfile>...

```

Run with `--help` to see which options are available.

Context files provided with `--context` are parsed and cached, so subsequent runs with the same file are faster. The cache for `foo/bar.c` is stored in `foo/bar.m2c`. These files can be ignored (added to `.gitignore`), and are automatically regenerated if context files change. Caching can be disabled with the `--no-cache` argument.

Multiple functions

By default, `mips\_to\_c` decompiles all functions in the text sections from the input assembly files.`mips\_to\_c` is able to perform a small amount of cross-function type inference, if the functions call each other.You can limit the function(s) that decompiled by providing the `-f <function name>` flags (or the "Function" dropdown on the website).

Global Declarations & Initializers

When provided input files with `data`, `rodata`, and/or `bss` sections, `mips\_to\_c` can generate the initializers for variables it knows the types of.Qualifier hints such as `const`, `static`, and `extern` are based on which sections the symbols appear in, or if they aren't provided at all.The output also includes prototypes for functions not declared in the context.`mips\_to\_c` cannot generate initializers for structs with bitfields (e.g. `unsigned foo: 3;`) or for symbols that it cannot infer the type of.For the latter, you can provide a type for the symbol the context.This feature is controlled with the `--globals` option (or "Global declarations" on the website):- `--globals=used` is the default behavior, global declarations are emitted for referenced symbols. Initializers are generated when the data/rodata sections are provided.- `--globals=none` disables globals entirely; only function definitions are emitted.- `--globals=all` includes all of the output in `used`, but also includes initializers for unreferenced symbols. This can be used to convert data/rodata files without decompiling any functions.

Struct Field Inference

By default, `mips\_to\_c` can use type information from decompiled functions to help fill in unknown struct fields.

This behavior can be disabled with `--no-unk-inference` ("Disable unknown struct/type inference" on the website).For structs in the context, the following fields treated as "unknown" space that can be inferred:

- `char` arrays with a name starting with `unk\_`, e.g. `char unk\_10[4];`

- any field with a type that starts with `UNK\_` or `MIPS2C\_UNK`, e.g. `UNK\_TYPE4 foo;`

Currently, struct field inference only works on structs without bitfields or [unnamed union fields](https://gcc.gnu.org/onlinedocs/gcc/Unnamed-Fields.html).

The output will include declarations for any struct with at least one inferred field.

Specifying stack variables

By default, `mips\_to\_c` infers the types of stack (local) variables, and names them with the `sp` prefix based on their offset.

Internally, the stack is represented as a struct, so it is possible to manually specify the names & types of stack variables by providing a struct declaration in the context. `mips\_to\_c` looks in the context for a struct with the tag name `\_mips2c\_stack\_<function name>` (e.g. `struct \_mips2c\_stack\_test` for a function `test()`).The size of the stack must exactly match the detected frame size, or `mips\_to\_c` will return an error.If you run `mips\_to\_c` with the `--stack-structs` option ("Stack struct templates" on the website), the output will include the inferred stack declaration, which can then be edited and provided as context by re-running `mips\_to\_c`.

Example

Here is an example for specifying the stack for the `custom\_stack` end-to-end test.First, run `mips\_to\_c` with the `--stack-structs` option to get the inferred struct for the `test()` function:

<details>

<summary><code>python3 mips\_to\_c.py tests/end\_to\_end/custom\_stack/irix-o2.s -f test --stack-structs</code></summary>

```c

struct \_mips2c\_stack\_test {

/\* 0x00 \*/ char pad0[0x20];

/\* 0x20 \*/ s8 sp20; /\* inferred \*/

/\* 0x21 \*/ char pad21[0x3]; /\* maybe part of sp20[4]? \*/

/\* 0x24 \*/ s32 sp24; /\* inferred \*/

/\* 0x28 \*/ s32 sp28; /\* inferred \*/

/\* 0x2C \*/ s8 sp2C; /\* inferred \*/

/\* 0x2D \*/ char pad2D[0x3]; /\* maybe part of sp2C[4]? \*/

/\* 0x30 \*/ s8 sp30; /\* inferred \*/

/\* 0x31 \*/ char pad31[0x3]; /\* maybe part of sp30[4]? \*/

/\* 0x34 \*/ s8 sp34; /\* inferred \*/

/\* 0x35 \*/ char pad35[0x2]; /\* maybe part of sp34[3]? \*/

/\* 0x37 \*/ s8 sp37; /\* inferred \*/

}; /\* size = 0x38 \*/

? func\_00400090(s8 \*); /\* static \*/

s32 test(void \*arg0); /\* static \*/

s32 test(void \*arg0) {

s8 sp37;

s8 sp34;

s8 sp30;

s8 sp2C;

s32 sp28;

s32 sp24;

s8 sp20;

s32 temp\_t4;

func\_00400090(&sp37);

func\_00400090(&sp34);

func\_00400090(&sp30);

func\_00400090(&sp2C);

func\_00400090(&sp20);

sp37 = arg0->unk0 + arg0->unk4;

sp34 = arg0->unk0 + arg0->unk8;

temp\_t4 = arg0->unk4 + arg0->unk8;

sp30 = temp\_t4;

sp20 = arg0->unk0 \* sp37;

sp24 = arg0->unk4 \* (s16) sp34;

sp28 = arg0->unk8 \* temp\_t4;

if (sp37 != 0) {

sp2C = arg0;

} else {

sp2C = &sp20;

}

return sp37 + (s16) sp34 + (s32) sp30 + \*(s32 \*) sp2C + sp24;

}

```

</details>

Now, based on the body of the `test()` function, we can make some guesses about the types of these variables, and give them more descriptive names:

```c

// Save this file as `test\_context.c`

struct Vec {

s32 x, y, z;

};

struct \_mips2c\_stack\_test {

char pad0[0x20];

struct Vec vec;

struct Vec \*vec\_ptr;

s32 scale\_z;

s16 scale\_y;

char pad36[1];

s8 scale\_x;

}; /\* size 0x38 \*/

int test(struct Vec \*vec\_arg);

```

Finally, re-run `mips\_to\_c` with our custom stack as part of the `--context`. The `--context` option can be specified multiple times to combine files.

<details>

<summary><code>python3 mips\_to\_c.py tests/end\_to\_end/custom\_stack/irix-o2.s -f test --context test\_context.c</code></summary>

```c

? func\_00400090(s8 \*); /\* static \*/

s32 test(struct Vec \*vec\_arg) {

s8 scale\_x;

s16 scale\_y;

s32 scale\_z;

struct Vec \*vec\_ptr;

struct Vec vec;

s32 temp\_t4;

func\_00400090(&scale\_x);

func\_00400090((s8 \*) &scale\_y);

func\_00400090((s8 \*) &scale\_z);

func\_00400090((s8 \*) &vec\_ptr);

func\_00400090((s8 \*) &vec);

scale\_x = vec\_arg->x + vec\_arg->y;

scale\_y = vec\_arg->x + vec\_arg->z;

temp\_t4 = vec\_arg->y + vec\_arg->z;

scale\_z = temp\_t4;

vec = vec\_arg->x \* scale\_x;

vec.y = vec\_arg->y \* scale\_y;

vec.z = vec\_arg->z \* temp\_t4;

if (scale\_x != 0) {

vec\_ptr = vec\_arg;

} else {

vec\_ptr = &vec;

}

return scale\_x + scale\_y + scale\_z + vec\_ptr->x + vec.y;

}

```

</details>

### Formatting

The following options control the formatting details of the output, such as braces style or numeric format. See `./mips\_to\_c.py --help` for more details.

(The option name on the website, if available, is in parentheses.)

- `--valid-syntax`

- `--allman` ("Allman braces")

- `--pointer-style` ("`\*` to the left")

- `--unk-underscore`

- `--hex-case`

- `--comment-style {multiline,oneline,none}` ("Comment style")

- `--comment-column N` ("Comment style")

- `--no-casts`

Note: `--valid-syntax` is used to produce output that is less human-readable, but is likely to directly compile without edits. This can be used to go directly from assembly to the permuter without human intervention.

### Debugging poor results (Advanced)

There are several options to `mips\_to\_c` which can be used to troubleshoot poor results. Many of these options produce more "primitive" output or debugging information.

- `--no-andor` ("Disable &&/||"): Disable complex conditional detection, such as `if (a && b)`. Instead, emit each part of the conditional as a separate `if` statement. Ands, ors, nots, etc. are usually represented with `goto`s.

- `--no-switches` ("Disable irregular switch detection"): Disable "irregular" `switch` statements, where the compiler emits a single `switch` as a series of branches and/or jump tables. By default, these are coalesced into a single `switch` and marked with an `/\* irregular \*/` comment.

- `--no-unk-inference` ("Disable unknown struct/type inference"): Disable attempting to infer struct fields/types in unknown struct sections and global symbols. See the [\_Struct Field Inference\_](#struct-field-inference) section above.

- `--gotos-only` ("Use gotos for everything"): Do not detect loops or complex conditionals. This format is close to a 1-1 translation of the assembly.

- Note: to use a goto for a single branch, don't use this flag, but add `# GOTO` to the assembly input.

- `--debug` ("Debug info"): include debug information inline with the code, such as basic block boundaries & labels.

- `--void` ("Force void return type"): assume that the decompiled function has return type `void`. Alternatively: provide the function prototype in the context.

#### Visualization

`mips\_to\_c` can generate an SVG representation of the control flow of a function, which can sometimes be helpful to untangle complex loops or early returns.

Pass `--visualize` on the command line, or use the "Visualize" button on the website. The output will be an SVG file.

Example to produce `my\_fn.svg` of `my\_fn()`:

```sh

python3 ./mips\_to\_c.py --visualize --context ctx.c -f my\_fn my\_asm.s > my\_fn.svg

```

## Contributing

There is much low-hanging fruit still. Take a look at the issues if you want to help out.

We use `black` to auto-format our code and `mypy` for type checking. We recommend using `pre-commit` to ensure only auto-formatted code is committed. To set these up, run:

```bash

pip install pre-commit black mypy

pre-commit install

```

Your commits will then be automatically formatted per commit. You can also manually run `black` on the command-line.

Type annotations are used for all Python code. `mypy` should pass without any errors.

To get pretty graph visualizations, install `graphviz` using `pip` and globally on your system (e.g. `sudo apt install graphviz`), and pass the `--visualize` flag.

## Tests

There is a small test suite, which works as follows:

- As you develop your commit, occasionally run `./run\_tests.py` to see if any tests have changed output.

These tests run the decompiler on a small corpus of IDO 5.3-compiled MIPS assembly.

- Before pushing your commit, run `./run\_tests.py --overwrite` to write changed tests to disk, and commit resultant changes.

### Running Decompilation Project Tests

It's possible to use the entire corpus of assembly files from decompilation projects as regression tests.

For now, the output of these tests are not tracked in version control.

You need to run `./run\_tests.py --overwrite ...` \*\*before\*\* making any code changes to create the baseline output.

As an example, if you have the `oot` project cloned locally in the parent directory containing `mips\_to\_c`, the following will decompile all of its assembly files.

```bash

./run\_tests.py --project ../oot --project-with-context ../oot

```

- Need to use the `ver/us` or `ver/jp` subfolder, e.g. `--project ../papermario/ver/us`

#### Creating Context Files

The following bash can be used in each decompilation project to create a "universal" context file that can be used for decompiling any assembly file in the project.

This creates `ctx.c` in the project directory.

```bash

cd mm # Or oot, papermario, etc.

find include/ src/ -type f -name "\*.h" | sed -e 's/.\*/#include "\0"/' > ctx\_includes.c

tools/m2ctx.py ctx\_includes.c

```

#### Notes for Majora's Mask

The build system in the MM decompilation project is currently being re-written.

It uses "transient" assembly that is not checked in, and in the normal build process it re-groups `.rodata` sections by function.

To use the MM project, run the following to \*just\* build the transient assembly files (and avoid running `split\_asm.py`).

```bash

cd mm

make distclean

make setup

make asm/disasm.dep

```

The repository should be setup correctly if there are `asm/code`, `asm/boot`, and `asm/overlays` folders with `.asm` files, but there \*should not\* be an `asm/non\_matchings` folder.

### Adding an End-to-End Test

You are encouraged to add new end-to-end tests using the `./tests/add\_test.py` script.

You'll need the IDO `cc` compiler and [sm64tools](https://github.com/queueRAM/sm64tools).

A good reference test to base your new test on is [`array-access`](tests/end\_to\_end/array-access).

Create a new directory in `tests/end\_to\_end`, and write the `orig.c` test case.

If you want the test to pass in C context, also add `irix-o2-flags.txt` & `irix-g-flags.txt` files.

After writing these files, run `add\_test.py` with the path to the new `orig.c` file, as shown below.

This example assumes that sm64tools is cloned & built in your home directory, and that the IDO compiler is available from the OOT decompilation project.

You should change these exported paths to match your system.

```bash

export SM64\_TOOLS=$HOME/sm64tools/build/

export IDO\_CC=$HOME/oot/tools/ido\_recomp/linux/7.1/cc

./tests/add\_test.py $PWD/tests/end\_to\_end/my-new-test/orig.c

```

This should create `irix-o2.s` and `irix-g.s` files in your test directory.

Now, run `./run\_tests.py --overwrite` to invoke the decompiler and write the output to `irix-o2-out.c` and `irix-g-out.c`.

Finally, `git add` your test to track it.

```bash

./run\_tests.py --overwrite

git add tests/end\_to\_end/my-new-test

CODE TRANSLATION

Main.s

    .arch armv6

    .eabi\_attribute 28, 1

    .eabi\_attribute 20, 1

    .eabi\_attribute 21, 1

    .eabi\_attribute 23, 3

    .eabi\_attribute 24, 1

    .eabi\_attribute 25, 1

    .eabi\_attribute 26, 2

    .eabi\_attribute 30, 6

    .eabi\_attribute 34, 1

    .eabi\_attribute 18, 4

    .file   "main.c"

    .section    .rodata

    .align  2

.LC0:

    .ascii  "w\000"

    .align  2

.LC1:

    .ascii  "Battleship.log\000"

    .align  2

.LC2:

    .ascii  "Pause\000"

    .align  2

.LC3:

    .ascii  "cls\000"

    .align  2

.LC4:

    .ascii  "\011 \*\*\*\* Enemy's Board \*\*\*\*\000"

    .align  2

.LC5:

    .ascii  "\012\011   \*\*\*\* Your Board \*\*\*\*\000"

    .align  2

.LC6:

    .ascii  "Hits: %d\011Shots: %d\012\000"

    .align  2

.LC7:

    .ascii  "Hits: %d\011Shots: %d   Target(%d,%d)\000"

    .align  2

.LC8:

    .ascii  "\011Hit!\000"

    .align  2

.LC9:

    .ascii  "\011Miss...\000"

    .align  2

.LC10:

    .ascii  "Hits: %d\011Shots: %d   Target: (%d,%d)\000"

    .align  2

.LC11:

    .ascii  "Hits: %d\011Score: %d\012\000"

    .align  2

.LC12:

    .ascii  "\012\011 \*\*\*\* Your Board \*\*\*\*\000"

    .align  2

.LC13:

    .ascii  "\012Player1 Wins!!!\012\000"

    .align  2

.LC14:

    .ascii  "\012Player2 Wins!!!\012\000"

    .align  2

.LC15:

    .ascii  "\*\*\*Player1 Stats\*\*\*\012Hits: %d\012Misses: %d\012Sh"

    .ascii  "ots: %d\012Accuracy: %.2lf%%\012\012\000"

    .align  2

.LC16:

    .ascii  "\*\*\*Player2 Stats\*\*\*\012Hits: %d\012Misses: %d\012Sh"

    .ascii  "ots: %d\012Accuracy: %.2lf%%\012\012\000"

    .text

    .align  2

    .global main

    .syntax unified

    .arm

    .fpu vfp

    .type   main, %function

main:

    @ args = 0, pretend = 0, frame = 536

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {r4, fp, lr}

    add fp, sp, #8

    sub sp, sp, #572

    mov r3, #0

    str r3, [fp, #-28]

    mov r3, #0

    str r3, [fp, #-132]

    mov r3, #0

    str r3, [fp, #-136]

    mov r3, #0

    str r3, [fp, #-140]

    mov r3, #0

    str r3, [fp, #-144]

    mov r3, #0

    str r3, [fp, #-32]

    mov r3, #5

    str r3, [fp, #-36]

    mov r3, #4

    str r3, [fp, #-40]

    mov r3, #3

    str r3, [fp, #-44]

    mov r3, #3

    str r3, [fp, #-48]

    mov r3, #2

    str r3, [fp, #-52]

    mov r3, #0

    str r3, [fp, #-56]

    mov r3, #0

    str r3, [fp, #-60]

    mov r3, #0

    str r3, [fp, #-16]

    mov r3, #0

    str r3, [fp, #-20]

    mov r3, #0

    str r3, [fp, #-64]

    mov r3, #0

    str r3, [fp, #-68]

    mov r3, #0

    str r3, [fp, #-148]

    mov r3, #0

    str r3, [fp, #-152]

    mvn r3, #0

    str r3, [fp, #-156]

    mvn r3, #0

    str r3, [fp, #-160]

    mov r3, #0

    str r3, [fp, #-164]

    mov r3, #0

    str r3, [fp, #-168]

    mov r3, #0

    str r3, [fp, #-172]

    mov r3, #0

    str r3, [fp, #-176]

    mov r3, #0

    str r3, [fp, #-180]

    mov r3, #0

    str r3, [fp, #-72]

    mov r3, #0

    str r3, [fp, #-76]

    mov r3, #0

    str r3, [fp, #-24]

    mov r3, #0

    str r3, [fp, #-80]

    mov r3, #0

    str r3, [fp, #-84]

    mov r3, #0

    str r3, [fp, #-88]

    mov r3, #0

    str r3, [fp, #-92]

    mov r3, #0

    str r3, [fp, #-96]

    mov r3, #0

    str r3, [fp, #-100]

    mov r3, #99

    strb    r3, [fp, #-101]

    mov r3, #98

    strb    r3, [fp, #-102]

    mov r3, #114

    strb    r3, [fp, #-103]

    mov r3, #115

    strb    r3, [fp, #-104]

    mov r3, #100

    strb    r3, [fp, #-105]

    mov r3, #0

    strb    r3, [fp, #-106]

    mov r3, #0

    strb    r3, [fp, #-107]

    mov r2, #0

    mov r3, #0

    strd    r2, [fp, #-116]

    mov r2, #0

    mov r3, #0

    strd    r2, [fp, #-124]

    mov r3, #0

    str r3, [fp, #-128]

    mov r0, #0

    bl  time

    mov r3, r0

    mov r0, r3

    bl  srand

    ldr r1, .L25

    ldr r0, .L25+4

    bl  fopen

    str r0, [fp, #-128]

    bl  welcome\_screen

    ldr r0, .L25+44

    bl  system

    ldr r0, .L25+48

    bl  system

    bl  deploy\_ships

    str r0, [fp, #-32]

    ldr r0, .L25+20

    bl  puts

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #444

    str r3, [sp, #4]

    ldr r3, [fp, #-36]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-101] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #444

    str r3, [sp, #4]

    ldr r3, [fp, #-40]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-102] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #444

    str r3, [sp, #4]

    ldr r3, [fp, #-44]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-103] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #444

    str r3, [sp, #4]

    ldr r3, [fp, #-48]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-104] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #444

    str r3, [sp, #4]

    ldr r3, [fp, #-52]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-105] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    ldr r0, .L25+16

    bl  puts

    ldr r3, [fp, #-32]

    cmp r3, #1

    bne .L2

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-36]

    str r3, [sp]

    mov r3, ip

    bl  manually\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-101] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-40]

    str r3, [sp]

    mov r3, ip

    bl  manually\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-102] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-44]

    str r3, [sp]

    mov r3, ip

    bl  manually\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-103] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-48]

    str r3, [sp]

    mov r3, ip

    bl  manually\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-104] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-52]

    str r3, [sp]

    mov r3, ip

    bl  manually\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-105] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

.L2:

    ldr r3, [fp, #-32]

    cmp r3, #2

    bne .L3

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-36]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-101] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-40]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-102] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-44]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-103] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-48]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-104] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #0

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    sub ip, fp, #144

    sub r2, fp, #140

    sub r1, fp, #136

    sub r0, fp, #132

    sub r3, fp, #344

    str r3, [sp, #4]

    ldr r3, [fp, #-52]

    str r3, [sp]

    mov r3, ip

    bl  randomly\_place\_ships\_on\_board

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-105] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

.L3:

    mvn r3, #0

    str r3, [fp, #-132]

    mvn r3, #0

    str r3, [fp, #-136]

    mvn r3, #0

    str r3, [fp, #-140]

    mvn r3, #0

    str r3, [fp, #-144]

    bl  select\_who\_starts\_first

    str r0, [fp, #-68]

    ldr r0, .L25+44

    bl  system

    ldr r3, [fp, #-68]

    cmp r3, #2

    bne .L4

    mov r3, #1

    str r3, [fp, #-20]

.L4:

    ldr r3, [fp, #-68]

    cmp r3, #1

    beq .L6

    cmp r3, #2

    beq .L7

    b   .L5

.L6:

    mvn r3, #0

    str r3, [fp, #-156]

    mvn r3, #0

    str r3, [fp, #-160]

    ldr r0, .L25+48

    bl  system

    ldr r0, .L25+20

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-152]

    ldr r2, [fp, #-20]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    ldr r0, .L25+16

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-148]

    ldr r2, [fp, #-16]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    sub r2, fp, #444

    sub r1, fp, #160

    sub r3, fp, #156

    mov r0, r3

    bl  shoot\_manual

    ldr r3, [fp, #-16]

    add r3, r3, #1

    str r3, [fp, #-16]

    ldr r0, .L25+48

    bl  system

    ldr r0, .L25+20

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    ldr r1, [fp, #-152]

    ldr r2, [fp, #-160]

    ldr r3, [fp, #-156]

    str r3, [sp]

    mov r3, r2

    ldr r2, [fp, #-20]

    ldr r0, .L25+8

    bl  printf

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #432

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #42

    bne .L8

    ldr r0, .L25+28

    bl  puts

    mov r3, #1

    str r3, [fp, #-24]

.L8:

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #432

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    bne .L9

    ldr r0, .L25+32

    bl  puts

    mov r3, #0

    str r3, [fp, #-24]

.L9:

    sub ip, fp, #172

    sub r2, fp, #168

    sub r1, fp, #164

    sub r0, fp, #444

    sub r3, fp, #180

    str r3, [sp, #4]

    sub r3, fp, #176

    str r3, [sp]

    mov r3, ip

    bl  check\_ship

    ldr r1, [fp, #-156]

    ldr r2, [fp, #-160]

    sub r3, fp, #180

    str r3, [sp, #20]

    sub r3, fp, #176

    str r3, [sp, #16]

    sub r3, fp, #172

    str r3, [sp, #12]

    sub r3, fp, #168

    str r3, [sp, #8]

    sub r3, fp, #164

    str r3, [sp, #4]

    ldr r3, [fp, #-24]

    str r3, [sp]

    mov r3, #1

    ldr r0, [fp, #-128]

    bl  output\_move

    mvn r3, #0

    str r3, [fp, #-156]

    mvn r3, #0

    str r3, [fp, #-160]

    ldr r0, .L25+16

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-148]

    ldr r2, [fp, #-16]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    ldr r0, .L25+44

    bl  system

    ldr r0, .L25+48

    bl  system

    ldr r3, [fp, #-148]

    cmp r3, #16

    bgt .L10

    ldr r0, .L25+20

    bl  puts

    mvn r3, #0

    str r3, [fp, #-156]

    mvn r3, #0

    str r3, [fp, #-160]

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-152]

    ldr r2, [fp, #-20]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    ldr r0, .L25+16

    bl  puts

    sub r2, fp, #344

    sub r1, fp, #160

    sub r3, fp, #156

    mov r0, r3

    bl  shoot\_random

    ldr r3, [fp, #-20]

    add r3, r3, #1

    str r3, [fp, #-20]

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    ldr r1, [fp, #-148]

    ldr r2, [fp, #-160]

    ldr r3, [fp, #-156]

    str r3, [sp]

    mov r3, r2

    ldr r2, [fp, #-16]

    ldr r0, .L25+24

    bl  printf

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #332

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #42

    bne .L11

    ldr r0, .L25+28

    bl  puts

    mov r3, #1

    str r3, [fp, #-24]

.L11:

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #332

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    bne .L12

    ldr r0, .L25+32

    bl  puts

    mov r3, #0

    str r3, [fp, #-24]

.L12:

    sub ip, fp, #172

    sub r2, fp, #168

    sub r1, fp, #164

    sub r0, fp, #344

    sub r3, fp, #180

    str r3, [sp, #4]

    sub r3, fp, #176

    str r3, [sp]

    mov r3, ip

    bl  check\_ship

    ldr r1, [fp, #-156]

    ldr r2, [fp, #-160]

    sub r3, fp, #180

    str r3, [sp, #20]

    sub r3, fp, #176

    str r3, [sp, #16]

    sub r3, fp, #172

    str r3, [sp, #12]

    sub r3, fp, #168

    str r3, [sp, #8]

    sub r3, fp, #164

    str r3, [sp, #4]

    ldr r3, [fp, #-24]

    str r3, [sp]

    mov r3, #2

    ldr r0, [fp, #-128]

    bl  output\_move

.L10:

    ldr r0, .L25+44

    bl  system

    ldr r0, .L25+48

    bl  system

    ldr r3, [fp, #-148]

    cmp r3, #16

    bgt .L23

    ldr r3, [fp, #-152]

    cmp r3, #16

    ble .L6

    b   .L23

.L26:

    .align  2

.L25:

    .word   .LC0

    .word   .LC1

    .word   .LC7

    .word   .LC11

    .word   .LC5

    .word   .LC4

    .word   .LC10

    .word   .LC8

    .word   .LC9

    .word   .LC12

    .word   .LC6

    .word   .LC2

    .word   .LC3

    .word   .LC13

    .word   .LC14

.L7:

    mvn r3, #0

    str r3, [fp, #-156]

    mvn r3, #0

    str r3, [fp, #-160]

    ldr r0, .L25+48

    bl  system

    ldr r0, .L25+20

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-152]

    ldr r2, [fp, #-20]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    ldr r0, .L25+16

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-148]

    ldr r2, [fp, #-16]

    mov r1, r3

    ldr r0, .L25+12

    bl  printf

    ldr r0, .L25+48

    bl  system

    ldr r0, .L25+20

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-152]

    ldr r2, [fp, #-20]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    ldr r0, .L25+16

    bl  puts

    sub r2, fp, #344

    sub r1, fp, #160

    sub r3, fp, #156

    mov r0, r3

    bl  shoot\_random

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-234]

    ldr r1, [fp, #-148]

    ldr r2, [fp, #-160]

    ldr r3, [fp, #-156]

    str r3, [sp]

    mov r3, r2

    ldr r2, [fp, #-16]

    ldr r0, .L25+24

    bl  printf

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #332

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #42

    bne .L14

    ldr r0, .L25+28

    bl  puts

    mov r3, #1

    str r3, [fp, #-24]

.L14:

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #332

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    bne .L15

    ldr r0, .L25+32

    bl  puts

    mov r3, #0

    str r3, [fp, #-24]

.L15:

    sub ip, fp, #172

    sub r2, fp, #168

    sub r1, fp, #164

    sub r0, fp, #344

    sub r3, fp, #180

    str r3, [sp, #4]

    sub r3, fp, #176

    str r3, [sp]

    mov r3, ip

    bl  check\_ship

    ldr r1, [fp, #-156]

    ldr r2, [fp, #-160]

    sub r3, fp, #180

    str r3, [sp, #20]

    sub r3, fp, #176

    str r3, [sp, #16]

    sub r3, fp, #172

    str r3, [sp, #12]

    sub r3, fp, #168

    str r3, [sp, #8]

    sub r3, fp, #164

    str r3, [sp, #4]

    ldr r3, [fp, #-24]

    str r3, [sp]

    mov r3, #2

    ldr r0, [fp, #-128]

    bl  output\_move

    ldr r3, [fp, #-152]

    cmp r3, #16

    bgt .L16

    sub r2, fp, #444

    sub r1, fp, #160

    sub r3, fp, #156

    mov r0, r3

    bl  shoot\_manual

    ldr r3, [fp, #-16]

    add r3, r3, #1

    str r3, [fp, #-16]

    ldr r0, .L25+48

    bl  system

    ldr r0, .L25+20

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #148

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #2

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #444

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    mov r3, r0

    strb    r3, [fp, #-334]

    ldr r1, [fp, #-152]

    ldr r2, [fp, #-160]

    ldr r3, [fp, #-156]

    str r3, [sp]

    mov r3, r2

    ldr r2, [fp, #-20]

    ldr r0, .L25+24

    bl  printf

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #432

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #42

    bne .L17

    ldr r0, .L25+28

    bl  puts

    mov r3, #1

    str r3, [fp, #-24]

.L17:

    ldr r2, [fp, #-156]

    ldr r1, [fp, #-160]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, fp, #12

    add r3, r2, r3

    add r3, r3, r1

    sub r3, r3, #432

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    bne .L18

    ldr r0, .L25+32

    bl  puts

    mov r3, #0

    str r3, [fp, #-24]

.L18:

    sub ip, fp, #172

    sub r2, fp, #168

    sub r1, fp, #164

    sub r0, fp, #444

    sub r3, fp, #180

    str r3, [sp, #4]

    sub r3, fp, #176

    str r3, [sp]

    mov r3, ip

    bl  check\_ship

    ldr r0, .L25+36

    bl  puts

    ldr r0, [fp, #-132]

    ldr ip, [fp, #-136]

    ldr lr, [fp, #-140]

    ldr r4, [fp, #-144]

    ldr r3, [fp, #-156]

    ldr r2, [fp, #-160]

    ldrb    r1, [fp, #-106] @ zero\_extendqisi2

    str r1, [sp, #28]

    sub r1, fp, #544

    str r1, [sp, #24]

    sub r1, fp, #152

    str r1, [sp, #20]

    str r2, [sp, #16]

    str r3, [sp, #12]

    mov r3, #1

    str r3, [sp, #8]

    mov r3, #1

    str r3, [sp, #4]

    sub r3, fp, #344

    str r3, [sp]

    mov r3, r4

    mov r2, lr

    mov r1, ip

    bl  initialize\_game\_board

    ldr r3, [fp, #-148]

    ldr r2, [fp, #-16]

    mov r1, r3

    ldr r0, .L25+40

    bl  printf

    ldr r1, [fp, #-156]

    ldr r2, [fp, #-160]

    sub r3, fp, #180

    str r3, [sp, #20]

    sub r3, fp, #176

    str r3, [sp, #16]

    sub r3, fp, #172

    str r3, [sp, #12]

    sub r3, fp, #168

    str r3, [sp, #8]

    sub r3, fp, #164

    str r3, [sp, #4]

    ldr r3, [fp, #-24]

    str r3, [sp]

    mov r3, #1

    ldr r0, [fp, #-128]

    bl  output\_move

.L16:

    ldr r3, [fp, #-20]

    add r3, r3, #1

    str r3, [fp, #-20]

    ldr r0, .L25+44

    bl  system

    ldr r0, .L25+48

    bl  system

    ldr r3, [fp, #-148]

    cmp r3, #16

    bgt .L24

    ldr r3, [fp, #-152]

    cmp r3, #16

    ble .L7

    b   .L24

.L23:

    nop

    b   .L5

.L24:

    nop

.L5:

    ldr r3, [fp, #-148]

    cmp r3, #17

    bne .L20

    ldr r0, .L25+52

    bl  puts

    mov r3, #1

    str r3, [fp, #-188]

    mov r3, #0

    str r3, [fp, #-220]

.L20:

    ldr r3, [fp, #-152]

    cmp r3, #17

    bne .L21

    ldr r0, .L25+56

    bl  puts

    mov r3, #0

    str r3, [fp, #-188]

    mov r3, #1

    str r3, [fp, #-220]

.L21:

    ldr r3, [fp, #-148]

    ldr r2, [fp, #-16]

    sub r3, r2, r3

    str r3, [fp, #-72]

    ldr r3, [fp, #-152]

    ldr r2, [fp, #-20]

    sub r3, r2, r3

    str r3, [fp, #-76]

    ldr r3, [fp, #-148]

    vmov    s15, r3 @ int

    vcvt.f64.s32    d7, s15

    vldr.64 d6, .L27

    vmul.f64    d5, d7, d6

    ldr r3, [fp, #-16]

    vmov    s15, r3 @ int

    vcvt.f64.s32    d6, s15

    vdiv.f64    d7, d5, d6

    vstr.64 d7, [fp, #-116]

    ldr r3, [fp, #-152]

    vmov    s15, r3 @ int

    vcvt.f64.s32    d7, s15

    vldr.64 d6, .L27

    vmul.f64    d5, d7, d6

    ldr r3, [fp, #-20]

    vmov    s15, r3 @ int

    vcvt.f64.s32    d6, s15

    vdiv.f64    d7, d5, d6

    vstr.64 d7, [fp, #-124]

    mov r3, #1

    str r3, [fp, #-212]

    ldr r3, [fp, #-148]

    str r3, [fp, #-208]

    ldr r3, [fp, #-72]

    str r3, [fp, #-204]

    ldr r3, [fp, #-16]

    str r3, [fp, #-200]

    ldrd    r2, [fp, #-116]

    strd    r2, [fp, #-196]

    mov r3, #2

    str r3, [fp, #-244]

    ldr r3, [fp, #-152]

    str r3, [fp, #-240]

    ldr r3, [fp, #-76]

    str r3, [fp, #-236]

    ldr r3, [fp, #-20]

    str r3, [fp, #-232]

    ldrd    r2, [fp, #-124]

    strd    r2, [fp, #-228]

    mov lr, sp

    sub ip, fp, #204

    ldmia   ip!, {r0, r1, r2, r3}

    stmia   lr!, {r0, r1, r2, r3}

    ldm ip, {r0, r1}

    stm lr, {r0, r1}

    sub r3, fp, #212

    ldm r3, {r2, r3}

    ldr r0, [fp, #-128]

    bl  output\_stats

    mov lr, sp

    sub ip, fp, #236

    ldmia   ip!, {r0, r1, r2, r3}

    stmia   lr!, {r0, r1, r2, r3}

    ldm ip, {r0, r1}

    stm lr, {r0, r1}

    sub r3, fp, #244

    ldm r3, {r2, r3}

    ldr r0, [fp, #-128]

    bl  output\_stats

    ldr r1, [fp, #-148]

    ldrd    r2, [fp, #-116]

    strd    r2, [sp]

    ldr r3, [fp, #-16]

    ldr r2, [fp, #-72]

    ldr r0, .L27+8

    bl  printf

    ldr r1, [fp, #-152]

    ldrd    r2, [fp, #-124]

    strd    r2, [sp]

    ldr r3, [fp, #-20]

    ldr r2, [fp, #-76]

    ldr r0, .L27+12

    bl  printf

    ldr r0, [fp, #-128]

    bl  fclose

    mov r3, #0

    mov r0, r3

    sub sp, fp, #8

    @ sp needed

    pop {r4, fp, pc}

.L28:

    .align  3

.L27:

    .word   0

    .word   1079574528

    .word   .LC15

    .word   .LC16

    .size   main, .-main

    .ident  "GCC: (Raspbian 6.3.0-18+rpi1) 6.3.0 20170516"

    .section    .note.GNU-stack,"",%progbits

Function.s

    .arch armv6

    .eabi\_attribute 28, 1

    .eabi\_attribute 20, 1

    .eabi\_attribute 21, 1

    .eabi\_attribute 23, 3

    .eabi\_attribute 24, 1

    .eabi\_attribute 25, 1

    .eabi\_attribute 26, 2

    .eabi\_attribute 30, 6

    .eabi\_attribute 34, 1

    .eabi\_attribute 18, 4

    .file   "function.c"

    .section    .rodata

    .align  2

.LC0:

    .ascii  "\012\011\011\011BATTLESHIP!!!\000"

    .align  2

.LC1:

    .ascii  "\012\011\011\011Rules of Battleship: \012\000"

    .align  2

.LC2:

    .ascii  "Players place their 'fleet' of 5 ships on their 'oc"

    .ascii  "ean', hidden from the\000"

    .align  2

.LC3:

    .ascii  "opponent's view. Taking turns, players call out the"

    .ascii  "ir 'shots' attempting to\000"

    .align  2

.LC4:

    .ascii  "get 'hits' on the opponent's ships in order to sink"

    .ascii  " them. Strategy and some\000"

    .align  2

.LC5:

    .ascii  "luck must be combined to be the first to locate and"

    .ascii  " sink all 5 opponent's\000"

    .align  2

.LC6:

    .ascii  "ships to win the game. The object of the game is to"

    .ascii  " be the first player to\000"

    .align  2

.LC7:

    .ascii  "sink all five of his opponent's ships. Each player "

    .ascii  "SECRETLY places his fleet\000"

    .align  2

.LC8:

    .ascii  "of 5 ships on his ocean grid. Ships may be placed i"

    .ascii  "n any horizontal or vertical\000"

    .align  2

.LC9:

    .ascii  "position - but NOT diagonally. You MAY NOT change t"

    .ascii  "he position of any ship.\000"

    .align  2

.LC10:

    .ascii  "To do so would be cheating!\012\000"

    .align  2

.LC11:

    .ascii  "This is a one player game where you are player1, an"

    .ascii  "d the computer is player2.\012\000"

    .align  2

.LC12:

    .ascii  "Types of Ships: \011Size:\012\000"

    .align  2

.LC13:

    .ascii  "Aircraft Carrier \0115 \012Battleship \011\0114 \012"

    .ascii  "Cruiser \011\0113 \012Submarine \011\0113 \012Destr"

    .ascii  "oyer \011\0112\012\000"

    .align  2

.LC14:

    .ascii  "Symbols: \012 c - Aircraft Carrier\012 b - Battlesh"

    .ascii  "ip\012 r - Cruiser\012 s - Submarine\012 d - Destro"

    .ascii  "yer\000"

    .align  2

.LC15:

    .ascii  "\012 ~ - Water\012 \* - Hit\012 m - Miss\000"

    .text

    .align  2

    .global welcome\_screen

    .syntax unified

    .arm

    .fpu vfp

    .type   welcome\_screen, %function

welcome\_screen:

    @ args = 0, pretend = 0, frame = 0

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    ldr r0, .L2

    bl  puts

    ldr r0, .L2+4

    bl  puts

    ldr r0, .L2+8

    bl  puts

    ldr r0, .L2+12

    bl  puts

    ldr r0, .L2+16

    bl  puts

    ldr r0, .L2+20

    bl  puts

    ldr r0, .L2+24

    bl  puts

    ldr r0, .L2+28

    bl  puts

    ldr r0, .L2+32

    bl  puts

    ldr r0, .L2+36

    bl  puts

    ldr r0, .L2+40

    bl  puts

    ldr r0, .L2+44

    bl  puts

    ldr r0, .L2+48

    bl  puts

    ldr r0, .L2+52

    bl  puts

    ldr r0, .L2+56

    bl  puts

    ldr r0, .L2+60

    bl  puts

    nop

    pop {fp, pc}

.L3:

    .align  2

.L2:

    .word   .LC0

    .word   .LC1

    .word   .LC2

    .word   .LC3

    .word   .LC4

    .word   .LC5

    .word   .LC6

    .word   .LC7

    .word   .LC8

    .word   .LC9

    .word   .LC10

    .word   .LC11

    .word   .LC12

    .word   .LC13

    .word   .LC14

    .word   .LC15

    .size   welcome\_screen, .-welcome\_screen

    .section    .rodata

    .align  2

.LC16:

    .ascii  "\012\011    0 1 2 3 4 5 6 7 8 9\000"

    .align  2

.LC17:

    .ascii  "\011 +---------------------+\012\011\000"

    .align  2

.LC18:

    .ascii  "%d | \000"

    .align  2

.LC19:

    .ascii  "%c \000"

    .align  2

.LC20:

    .ascii  "| %d\012\011\000"

    .align  2

.LC21:

    .ascii  " +---------------------+\000"

    .align  2

.LC22:

    .ascii  "\011    0 1 2 3 4 5 6 7 8 9\012\000"

    .text

    .align  2

    .global initialize\_game\_board

    .syntax unified

    .arm

    .fpu vfp

    .type   initialize\_game\_board, %function

initialize\_game\_board:

    @ args = 32, pretend = 0, frame = 24

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #24

    str r0, [fp, #-16]

    str r1, [fp, #-20]

    str r2, [fp, #-24]

    str r3, [fp, #-28]

    mov r3, #0

    str r3, [fp, #-8]

    mov r3, #0

    str r3, [fp, #-12]

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L5

    ldr r0, .L32

    bl  puts

    ldr r0, .L32+4

    bl  printf

.L5:

    mov r3, #0

    str r3, [fp, #-8]

    b   .L6

.L29:

    ldr r3, [fp, #-8]

    cmp r3, #8

    bgt .L7

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L7

    ldr r1, [fp, #-8]

    ldr r0, .L32+8

    bl  printf

.L7:

    ldr r3, [fp, #-8]

    cmp r3, #9

    bne .L8

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L8

    ldr r1, [fp, #-8]

    ldr r0, .L32+8

    bl  printf

.L8:

    mov r3, #0

    str r3, [fp, #-12]

    b   .L9

.L28:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #99

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #98

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #114

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #115

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #100

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #42

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #126

    strb    r2, [r3]

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-16]

    cmp r2, r3

    bne .L11

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-20]

    cmp r2, r3

    beq .L12

.L11:

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-24]

    cmp r2, r3

    bne .L13

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-28]

    cmp r2, r3

    bne .L13

.L12:

    ldr r3, [fp, #8]

    cmn r3, #1

    beq .L13

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r2, [fp, #32]

    strb    r2, [r3]

.L13:

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-16]

    cmp r2, r3

    bge .L14

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-24]

    cmp r2, r3

    ble .L14

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-20]

    cmp r2, r3

    beq .L15

.L14:

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-16]

    cmp r2, r3

    ble .L16

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-24]

    cmp r2, r3

    bge .L16

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-20]

    cmp r2, r3

    bne .L16

.L15:

    ldr r3, [fp, #8]

    cmn r3, #1

    beq .L16

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r2, [fp, #32]

    strb    r2, [r3]

.L16:

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-20]

    cmp r2, r3

    bge .L17

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-28]

    cmp r2, r3

    ble .L17

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-16]

    cmp r2, r3

    beq .L18

.L17:

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-20]

    cmp r2, r3

    ble .L10

    ldr r2, [fp, #-12]

    ldr r3, [fp, #-28]

    cmp r2, r3

    bge .L10

    ldr r2, [fp, #-8]

    ldr r3, [fp, #-16]

    cmp r2, r3

    bne .L10

.L18:

    ldr r3, [fp, #8]

    cmn r3, #1

    beq .L10

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r2, [fp, #32]

    strb    r2, [r3]

.L10:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #126

    bne .L19

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #126

    strb    r2, [r3]

.L19:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L20

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #115

    bhi .L20

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    beq .L20

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #126

    strb    r2, [r3]

.L20:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #109

    bne .L21

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #109

    strb    r2, [r3]

.L21:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #42

    bne .L22

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #42

    strb    r2, [r3]

.L22:

    ldr r2, [fp, #-8]

    ldr r3, [fp, #16]

    cmp r2, r3

    bne .L23

    ldr r2, [fp, #-12]

    ldr r3, [fp, #20]

    cmp r2, r3

    bne .L23

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L24

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #115

    bhi .L24

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #42

    strb    r2, [r3]

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #42

    strb    r2, [r3]

    ldr r3, [fp, #24]

    ldr r3, [r3]

    add r2, r3, #1

    ldr r3, [fp, #24]

    str r2, [r3]

.L24:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #126

    bne .L23

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #109

    strb    r2, [r3]

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    mov r2, #109

    strb    r2, [r3]

.L23:

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L25

    ldr r3, [fp, #12]

    cmp r3, #1

    bne .L25

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #4]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    mov r1, r3

    ldr r0, .L32+12

    bl  printf

.L25:

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L26

    ldr r3, [fp, #12]

    cmp r3, #2

    bne .L26

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #28]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    mov r1, r3

    ldr r0, .L32+12

    bl  printf

.L26:

    ldr r3, [fp, #-12]

    cmp r3, #9

    bne .L27

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L27

    ldr r1, [fp, #-8]

    ldr r0, .L32+16

    bl  printf

.L27:

    ldr r3, [fp, #-12]

    add r3, r3, #1

    str r3, [fp, #-12]

.L9:

    ldr r3, [fp, #-12]

    cmp r3, #9

    ble .L28

    ldr r3, [fp, #-8]

    add r3, r3, #1

    str r3, [fp, #-8]

.L6:

    ldr r3, [fp, #-8]

    cmp r3, #9

    ble .L29

    ldr r3, [fp, #8]

    cmp r3, #0

    beq .L30

    ldr r0, .L32+20

    bl  puts

    ldr r0, .L32+24

    bl  puts

.L30:

    ldr r3, [fp, #4]

    uxtb    r3, r3

    mov r0, r3

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L33:

    .align  2

.L32:

    .word   .LC16

    .word   .LC17

    .word   .LC18

    .word   .LC19

    .word   .LC20

    .word   .LC21

    .word   .LC22

    .size   initialize\_game\_board, .-initialize\_game\_board

    .section    .rodata

    .align  2

.LC23:

    .ascii  "Player1 (Human) goes first. . .\012\000"

    .align  2

.LC24:

    .ascii  "Player2 (CPU) goes first. . .\012\000"

    .text

    .align  2

    .global select\_who\_starts\_first

    .syntax unified

    .arm

    .fpu vfp

    .type   select\_who\_starts\_first, %function

select\_who\_starts\_first:

    @ args = 0, pretend = 0, frame = 8

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #8

    mov r3, #0

    str r3, [fp, #-12]

    mov r3, #0

    str r3, [fp, #-8]

    bl  rand

    mov r2, r0

    asr r3, r2, #31

    lsr r3, r3, #31

    add r2, r2, r3

    and r2, r2, #1

    sub r3, r2, r3

    str r3, [fp, #-12]

    ldr r3, [fp, #-12]

    cmp r3, #0

    bne .L35

    ldr r0, .L38

    bl  puts

    mov r3, #1

    str r3, [fp, #-8]

.L35:

    ldr r3, [fp, #-12]

    cmp r3, #1

    bne .L36

    ldr r0, .L38+4

    bl  puts

    mov r3, #2

    str r3, [fp, #-8]

.L36:

    ldr r3, [fp, #-8]

    mov r0, r3

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L39:

    .align  2

.L38:

    .word   .LC23

    .word   .LC24

    .size   select\_who\_starts\_first, .-select\_who\_starts\_first

    .section    .rodata

    .align  2

.LC25:

    .ascii  "How would you like to deploy your ships?\0121) Manu"

    .ascii  "ally\0122) Randomly\012\000"

    .align  2

.LC26:

    .ascii  "%d\000"

    .text

    .align  2

    .global deploy\_ships

    .syntax unified

    .arm

    .fpu vfp

    .type   deploy\_ships, %function

deploy\_ships:

    @ args = 0, pretend = 0, frame = 8

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #8

    mov r3, #0

    str r3, [fp, #-8]

    ldr r0, .L42

    bl  puts

    sub r3, fp, #8

    mov r1, r3

    ldr r0, .L42+4

    bl  \_\_isoc99\_scanf

    ldr r3, [fp, #-8]

    mov r0, r3

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L43:

    .align  2

.L42:

    .word   .LC25

    .word   .LC26

    .size   deploy\_ships, .-deploy\_ships

    .section    .rodata

    .align  2

.LC27:

    .ascii  "Enter the endpoint coordinates for your Ship, size:"

    .ascii  " %d: \012\000"

    .align  2

.LC28:

    .ascii  "%d %d %d %d\000"

    .align  2

.LC29:

    .ascii  "The ship cannot be diagonal!\000"

    .align  2

.LC30:

    .ascii  "The ship does not fit those coordinates!\000"

    .align  2

.LC31:

    .ascii  "Your ships cannot overlap!\000"

    .text

    .align  2

    .global manually\_place\_ships\_on\_board

    .syntax unified

    .arm

    .fpu vfp

    .type   manually\_place\_ships\_on\_board, %function

manually\_place\_ships\_on\_board:

    @ args = 8, pretend = 0, frame = 56

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #64

    str r0, [fp, #-48]

    str r1, [fp, #-52]

    str r2, [fp, #-56]

    str r3, [fp, #-60]

    mov r3, #0

    str r3, [fp, #-8]

    mov r3, #0

    str r3, [fp, #-12]

    mov r3, #0

    str r3, [fp, #-16]

    mov r3, #0

    str r3, [fp, #-20]

    mov r3, #0

    str r3, [fp, #-24]

    mov r3, #0

    str r3, [fp, #-28]

    mov r3, #0

    str r3, [fp, #-32]

    mov r3, #0

    str r3, [fp, #-36]

    mov r3, #0

    str r3, [fp, #-40]

.L49:

    mov r3, #0

    str r3, [fp, #-8]

    mov r3, #0

    str r3, [fp, #-12]

    mov r3, #0

    str r3, [fp, #-16]

    mov r3, #0

    str r3, [fp, #-20]

    mov r3, #0

    str r3, [fp, #-24]

    mov r3, #0

    str r3, [fp, #-28]

    mov r3, #0

    str r3, [fp, #-32]

    ldr r3, [fp, #-48]

    mov r2, #0

    str r2, [r3]

    ldr r3, [fp, #-52]

    mov r2, #0

    str r2, [r3]

    ldr r3, [fp, #-56]

    mov r2, #0

    str r2, [r3]

    ldr r3, [fp, #-60]

    mov r2, #0

    str r2, [r3]

    ldr r1, [fp, #4]

    ldr r0, .L77

    bl  printf

    ldr r3, [fp, #-56]

    str r3, [sp]

    ldr r3, [fp, #-60]

    ldr r2, [fp, #-48]

    ldr r1, [fp, #-52]

    ldr r0, .L77+4

    bl  \_\_isoc99\_scanf

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r2, r3

    beq .L45

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r2, r3

    beq .L45

    ldr r0, .L77+8

    bl  puts

.L45:

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r2, r3

    bne .L46

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    sub r3, r2, r3

    cmp r3, #0

    rsblt   r3, r3, #0

    add r3, r3, #1

    str r3, [fp, #-8]

.L46:

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r2, r3

    bne .L47

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    sub r3, r2, r3

    cmp r3, #0

    rsblt   r3, r3, #0

    add r3, r3, #1

    str r3, [fp, #-8]

.L47:

    ldr r2, [fp, #-8]

    ldr r3, [fp, #4]

    cmp r2, r3

    beq .L48

    ldr r0, .L77+12

    bl  puts

.L48:

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    cmp r3, #0

    blt .L49

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L49

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    cmp r3, #0

    bge .L50

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L49

.L50:

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r3, #0

    blt .L49

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L49

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r3, #0

    bge .L51

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L49

.L51:

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r2, r3

    beq .L52

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r2, r3

    bne .L49

.L52:

    ldr r2, [fp, #-8]

    ldr r3, [fp, #4]

    cmp r2, r3

    bne .L49

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r2, r3

    bne .L53

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r2, r3

    ble .L54

    ldr r3, [fp, #4]

    cmp r3, #2

    ble .L55

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    add r3, r3, #1

    str r3, [fp, #-12]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    str r3, [fp, #-16]

.L55:

    ldr r3, [fp, #4]

    cmp r3, #3

    ble .L56

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    add r3, r3, #2

    str r3, [fp, #-20]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    str r3, [fp, #-24]

.L56:

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L54

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    add r3, r3, #3

    str r3, [fp, #-28]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    str r3, [fp, #-32]

.L54:

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r2, r3

    bge .L53

    ldr r3, [fp, #4]

    cmp r3, #2

    ble .L57

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    add r3, r3, #1

    str r3, [fp, #-12]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    str r3, [fp, #-16]

.L57:

    ldr r3, [fp, #4]

    cmp r3, #3

    ble .L58

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    add r3, r3, #2

    str r3, [fp, #-20]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    str r3, [fp, #-24]

.L58:

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L53

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    add r3, r3, #3

    str r3, [fp, #-28]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    str r3, [fp, #-32]

.L53:

    ldr r3, [fp, #-48]

    ldr r2, [r3]

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    cmp r2, r3

    bne .L59

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r2, r3

    bge .L60

    ldr r3, [fp, #4]

    cmp r3, #2

    ble .L61

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    str r3, [fp, #-12]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    add r3, r3, #1

    str r3, [fp, #-16]

.L61:

    ldr r3, [fp, #4]

    cmp r3, #3

    ble .L62

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    str r3, [fp, #-20]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    add r3, r3, #2

    str r3, [fp, #-24]

.L62:

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L60

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    str r3, [fp, #-28]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    add r3, r3, #3

    str r3, [fp, #-32]

.L60:

    ldr r3, [fp, #-52]

    ldr r2, [r3]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    cmp r2, r3

    ble .L59

    ldr r3, [fp, #4]

    cmp r3, #2

    ble .L63

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    str r3, [fp, #-12]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    add r3, r3, #1

    str r3, [fp, #-16]

.L63:

    ldr r3, [fp, #4]

    cmp r3, #3

    ble .L64

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    str r3, [fp, #-20]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    add r3, r3, #2

    str r3, [fp, #-24]

.L64:

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L59

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    str r3, [fp, #-28]

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    add r3, r3, #3

    str r3, [fp, #-32]

.L59:

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L65

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L66

.L65:

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L67

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L66

.L67:

    ldr r2, [fp, #-12]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-16]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L68

    ldr r2, [fp, #-12]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-16]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bhi .L68

    ldr r3, [fp, #4]

    cmp r3, #2

    bgt .L66

.L68:

    ldr r2, [fp, #-20]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-24]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L69

    ldr r2, [fp, #-20]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-24]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bhi .L69

    ldr r3, [fp, #4]

    cmp r3, #3

    bgt .L66

.L69:

    ldr r2, [fp, #-28]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-32]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L70

    ldr r2, [fp, #-28]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-32]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bhi .L70

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L70

.L66:

    ldr r0, .L77+16

    bl  puts

.L70:

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L71

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L49

.L71:

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L72

    ldr r3, [fp, #-56]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-60]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L49

.L72:

    ldr r2, [fp, #-12]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-16]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L73

    ldr r2, [fp, #-12]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-16]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bhi .L73

    ldr r3, [fp, #4]

    cmp r3, #2

    bgt .L49

.L73:

    ldr r2, [fp, #-20]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-24]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L74

    ldr r2, [fp, #-20]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-24]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bhi .L74

    ldr r3, [fp, #4]

    cmp r3, #3

    bgt .L49

.L74:

    ldr r2, [fp, #-28]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-32]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L76

    ldr r2, [fp, #-28]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-32]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bhi .L76

    ldr r3, [fp, #4]

    cmp r3, #4

    bgt .L49

.L76:

    nop

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L78:

    .align  2

.L77:

    .word   .LC27

    .word   .LC28

    .word   .LC29

    .word   .LC30

    .word   .LC31

    .size   manually\_place\_ships\_on\_board, .-manually\_place\_ships\_on\_board

    .align  2

    .global randomly\_place\_ships\_on\_board

    .syntax unified

    .arm

    .fpu vfp

    .type   randomly\_place\_ships\_on\_board, %function

randomly\_place\_ships\_on\_board:

    @ args = 8, pretend = 0, frame = 48

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #48

    str r0, [fp, #-40]

    str r1, [fp, #-44]

    str r2, [fp, #-48]

    str r3, [fp, #-52]

    mov r3, #0

    str r3, [fp, #-32]

    mov r3, #0

    str r3, [fp, #-8]

    mov r3, #0

    str r3, [fp, #-12]

    mov r3, #0

    str r3, [fp, #-16]

    mov r3, #0

    str r3, [fp, #-20]

    mov r3, #0

    str r3, [fp, #-24]

    mov r3, #0

    str r3, [fp, #-28]

.L86:

    bl  rand

    mov r1, r0

    ldr r3, .L93

    smull   r2, r3, r3, r1

    asr r2, r3, #2

    asr r3, r1, #31

    sub r2, r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, r1, r3

    ldr r3, [fp, #-40]

    str r2, [r3]

    bl  rand

    mov r1, r0

    ldr r3, .L93

    smull   r2, r3, r3, r1

    asr r2, r3, #2

    asr r3, r1, #31

    sub r2, r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, r1, r3

    ldr r3, [fp, #-44]

    str r2, [r3]

    bl  rand

    mov r2, r0

    asr r3, r2, #31

    lsr r3, r3, #31

    add r2, r2, r3

    and r2, r2, #1

    sub r3, r2, r3

    str r3, [fp, #-32]

    ldr r3, [fp, #-32]

    cmp r3, #0

    bne .L80

    ldr r3, [fp, #-40]

    ldr r2, [r3]

    ldr r3, [fp, #4]

    sub r3, r2, r3

    add r2, r3, #1

    ldr r3, [fp, #-48]

    str r2, [r3]

    ldr r3, [fp, #-44]

    ldr r2, [r3]

    ldr r3, [fp, #-52]

    str r2, [r3]

    ldr r3, [fp, #4]

    cmp r3, #2

    ble .L81

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    add r3, r3, #1

    str r3, [fp, #-8]

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    str r3, [fp, #-12]

.L81:

    ldr r3, [fp, #4]

    cmp r3, #3

    ble .L82

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    add r3, r3, #2

    str r3, [fp, #-16]

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    str r3, [fp, #-20]

.L82:

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L80

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    add r3, r3, #3

    str r3, [fp, #-24]

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    str r3, [fp, #-28]

.L80:

    ldr r3, [fp, #-32]

    cmp r3, #1

    bne .L83

    ldr r3, [fp, #-44]

    ldr r2, [r3]

    ldr r3, [fp, #4]

    sub r3, r2, r3

    add r2, r3, #1

    ldr r3, [fp, #-52]

    str r2, [r3]

    ldr r3, [fp, #-40]

    ldr r2, [r3]

    ldr r3, [fp, #-48]

    str r2, [r3]

    ldr r3, [fp, #4]

    cmp r3, #2

    ble .L84

    ldr r3, [fp, #-40]

    ldr r3, [r3]

    str r3, [fp, #-8]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    add r3, r3, #1

    str r3, [fp, #-12]

.L84:

    ldr r3, [fp, #4]

    cmp r3, #3

    ble .L85

    ldr r3, [fp, #-40]

    ldr r3, [r3]

    str r3, [fp, #-16]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    add r3, r3, #2

    str r3, [fp, #-20]

.L85:

    ldr r3, [fp, #4]

    cmp r3, #4

    ble .L83

    ldr r3, [fp, #-40]

    ldr r3, [r3]

    str r3, [fp, #-24]

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    add r3, r3, #3

    str r3, [fp, #-28]

.L83:

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    cmp r3, #0

    ble .L86

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    cmp r3, #0

    ble .L86

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L86

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L86

    ldr r3, [fp, #-40]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L87

    ldr r3, [fp, #-40]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L86

.L87:

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L88

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L86

.L88:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L89

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L86

.L89:

    ldr r2, [fp, #-16]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-20]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L90

    ldr r2, [fp, #-16]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-20]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L86

.L90:

    ldr r2, [fp, #-24]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-28]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #97

    bls .L92

    ldr r2, [fp, #-24]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #8]

    add r2, r3, r2

    ldr r3, [fp, #-28]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #121

    bls .L86

.L92:

    nop

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L94:

    .align  2

.L93:

    .word   1717986919

    .size   randomly\_place\_ships\_on\_board, .-randomly\_place\_ships\_on\_board

    .section    .rodata

    .align  2

.LC32:

    .ascii  "Enter target coordinates: \000"

    .align  2

.LC33:

    .ascii  "%d %d\000"

    .align  2

.LC34:

    .ascii  "Your target coordinates must be from 0-9!\000"

    .align  2

.LC35:

    .ascii  "You have already shot there! \000"

    .text

    .align  2

    .global shoot\_manual

    .syntax unified

    .arm

    .fpu vfp

    .type   shoot\_manual, %function

shoot\_manual:

    @ args = 0, pretend = 0, frame = 16

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #16

    str r0, [fp, #-8]

    str r1, [fp, #-12]

    str r2, [fp, #-16]

.L98:

    ldr r0, .L101

    bl  puts

    ldr r2, [fp, #-8]

    ldr r1, [fp, #-12]

    ldr r0, .L101+4

    bl  \_\_isoc99\_scanf

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L96

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    cmp r3, #0

    blt .L96

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L96

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    cmp r3, #0

    bge .L97

.L96:

    ldr r0, .L101+8

    bl  puts

.L97:

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L98

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    cmp r3, #0

    blt .L98

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    cmp r3, #9

    bgt .L98

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    cmp r3, #0

    blt .L98

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-16]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #42

    beq .L99

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-16]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #109

    bne .L100

.L99:

    ldr r0, .L101+12

    bl  puts

.L100:

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-16]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #42

    beq .L98

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-16]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #109

    beq .L98

    nop

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L102:

    .align  2

.L101:

    .word   .LC32

    .word   .LC33

    .word   .LC34

    .word   .LC35

    .size   shoot\_manual, .-shoot\_manual

    .align  2

    .global shoot\_random

    .syntax unified

    .arm

    .fpu vfp

    .type   shoot\_random, %function

shoot\_random:

    @ args = 0, pretend = 0, frame = 16

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #16

    str r0, [fp, #-8]

    str r1, [fp, #-12]

    str r2, [fp, #-16]

.L104:

    bl  rand

    mov r1, r0

    ldr r3, .L105

    smull   r2, r3, r3, r1

    asr r2, r3, #2

    asr r3, r1, #31

    sub r2, r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, r1, r3

    ldr r3, [fp, #-8]

    str r2, [r3]

    bl  rand

    mov r1, r0

    ldr r3, .L105

    smull   r2, r3, r3, r1

    asr r2, r3, #2

    asr r3, r1, #31

    sub r2, r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    sub r2, r1, r3

    ldr r3, [fp, #-12]

    str r2, [r3]

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-16]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #42

    beq .L104

    ldr r3, [fp, #-8]

    ldr r3, [r3]

    mov r2, r3

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-16]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    ldr r3, [r3]

    ldrb    r3, [r2, r3]    @ zero\_extendqisi2

    cmp r3, #109

    beq .L104

    nop

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L106:

    .align  2

.L105:

    .word   1717986919

    .size   shoot\_random, .-shoot\_random

    .section    .rodata

    .align  2

.LC36:

    .ascii  "You sunk my Aircraft Carrier!\000"

    .align  2

.LC37:

    .ascii  "You sunk my Battleship!\000"

    .align  2

.LC38:

    .ascii  "You sunk my Cruiser!\000"

    .align  2

.LC39:

    .ascii  "You sunk my Submarine!\000"

    .align  2

.LC40:

    .ascii  "You sunk my Destroyer!\000"

    .text

    .align  2

    .global check\_ship

    .syntax unified

    .arm

    .fpu vfp

    .type   check\_ship, %function

check\_ship:

    @ args = 8, pretend = 0, frame = 48

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #48

    str r0, [fp, #-40]

    str r1, [fp, #-44]

    str r2, [fp, #-48]

    str r3, [fp, #-52]

    mov r3, #0

    str r3, [fp, #-8]

    mov r3, #0

    str r3, [fp, #-12]

    mov r3, #0

    str r3, [fp, #-16]

    mov r3, #0

    str r3, [fp, #-20]

    mov r3, #0

    str r3, [fp, #-24]

    mov r3, #0

    str r3, [fp, #-28]

    mov r3, #0

    str r3, [fp, #-32]

    mov r3, #0

    str r3, [fp, #-8]

    b   .L108

.L116:

    mov r3, #0

    str r3, [fp, #-12]

    b   .L109

.L115:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-40]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #99

    bne .L110

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    cmp r3, #0

    bne .L110

    ldr r3, [fp, #-16]

    add r3, r3, #1

    str r3, [fp, #-16]

.L110:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-40]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #98

    bne .L111

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    cmp r3, #0

    bne .L111

    ldr r3, [fp, #-20]

    add r3, r3, #1

    str r3, [fp, #-20]

.L111:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-40]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #114

    bne .L112

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    cmp r3, #0

    bne .L112

    ldr r3, [fp, #-24]

    add r3, r3, #1

    str r3, [fp, #-24]

.L112:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-40]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #115

    bne .L113

    ldr r3, [fp, #4]

    ldr r3, [r3]

    cmp r3, #0

    bne .L113

    ldr r3, [fp, #-28]

    add r3, r3, #1

    str r3, [fp, #-28]

.L113:

    ldr r2, [fp, #-8]

    mov r3, r2

    lsl r3, r3, #2

    add r3, r3, r2

    lsl r3, r3, #1

    mov r2, r3

    ldr r3, [fp, #-40]

    add r2, r3, r2

    ldr r3, [fp, #-12]

    add r3, r2, r3

    ldrb    r3, [r3]    @ zero\_extendqisi2

    cmp r3, #100

    bne .L114

    ldr r3, [fp, #8]

    ldr r3, [r3]

    cmp r3, #0

    bne .L114

    ldr r3, [fp, #-32]

    add r3, r3, #1

    str r3, [fp, #-32]

.L114:

    ldr r3, [fp, #-12]

    add r3, r3, #1

    str r3, [fp, #-12]

.L109:

    ldr r3, [fp, #-12]

    cmp r3, #9

    ble .L115

    ldr r3, [fp, #-8]

    add r3, r3, #1

    str r3, [fp, #-8]

.L108:

    ldr r3, [fp, #-8]

    cmp r3, #9

    ble .L116

    ldr r3, [fp, #-16]

    cmp r3, #0

    bne .L117

    ldr r3, [fp, #-44]

    ldr r3, [r3]

    cmn r3, #1

    beq .L117

    ldr r0, .L122

    bl  puts

    ldr r3, [fp, #-44]

    mov r2, #1

    str r2, [r3]

.L117:

    ldr r3, [fp, #-20]

    cmp r3, #0

    bne .L118

    ldr r3, [fp, #-48]

    ldr r3, [r3]

    cmn r3, #1

    beq .L118

    ldr r0, .L122+4

    bl  puts

    ldr r3, [fp, #-48]

    mov r2, #1

    str r2, [r3]

.L118:

    ldr r3, [fp, #-24]

    cmp r3, #0

    bne .L119

    ldr r3, [fp, #-52]

    ldr r3, [r3]

    cmn r3, #1

    beq .L119

    ldr r0, .L122+8

    bl  puts

    ldr r3, [fp, #-52]

    mov r2, #1

    str r2, [r3]

.L119:

    ldr r3, [fp, #-28]

    cmp r3, #0

    bne .L120

    ldr r3, [fp, #4]

    ldr r3, [r3]

    cmn r3, #1

    beq .L120

    ldr r0, .L122+12

    bl  puts

    ldr r3, [fp, #4]

    mov r2, #1

    str r2, [r3]

.L120:

    ldr r3, [fp, #-32]

    cmp r3, #0

    bne .L121

    ldr r3, [fp, #8]

    ldr r3, [r3]

    cmn r3, #1

    beq .L121

    ldr r0, .L122+16

    bl  puts

    ldr r3, [fp, #8]

    mov r2, #1

    str r2, [r3]

.L121:

    nop

    mov r0, r3

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L123:

    .align  2

.L122:

    .word   .LC36

    .word   .LC37

    .word   .LC38

    .word   .LC39

    .word   .LC40

    .size   check\_ship, .-check\_ship

    .section    .rodata

    .align  2

.LC41:

    .ascii  "Player%d: (%d,%d)\011\000"

    .align  2

.LC42:

    .ascii  "Hit!\012\000"

    .align  2

.LC43:

    .ascii  "Miss...\012\000"

    .align  2

.LC44:

    .ascii  "Sunk Aircraft Carrier!\012\000"

    .align  2

.LC45:

    .ascii  "Sunk Battleship!\012\000"

    .align  2

.LC46:

    .ascii  "Sunk Cruiser!\012\000"

    .align  2

.LC47:

    .ascii  "Sunk Submarine!\012\000"

    .align  2

.LC48:

    .ascii  "Sunk Destroyer!\012\000"

    .text

    .align  2

    .global output\_move

    .syntax unified

    .arm

    .fpu vfp

    .type   output\_move, %function

output\_move:

    @ args = 24, pretend = 0, frame = 16

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #24

    str r0, [fp, #-8]

    str r1, [fp, #-12]

    str r2, [fp, #-16]

    str r3, [fp, #-20]

    ldr r3, [fp, #-12]

    cmn r3, #1

    beq .L125

    ldr r3, [fp, #-16]

    cmn r3, #1

    beq .L125

    ldr r3, [fp, #-16]

    str r3, [sp]

    ldr r3, [fp, #-12]

    ldr r2, [fp, #-20]

    ldr r1, .L133

    ldr r0, [fp, #-8]

    bl  fprintf

.L125:

    ldr r3, [fp, #4]

    cmp r3, #1

    bne .L126

    ldr r3, [fp, #-8]

    mov r2, #5

    mov r1, #1

    ldr r0, .L133+4

    bl  fwrite

.L126:

    ldr r3, [fp, #4]

    cmp r3, #0

    bne .L127

    ldr r3, [fp, #-8]

    mov r2, #8

    mov r1, #1

    ldr r0, .L133+8

    bl  fwrite

.L127:

    ldr r3, [fp, #8]

    ldr r3, [r3]

    cmp r3, #1

    bne .L128

    ldr r3, [fp, #-8]

    mov r2, #23

    mov r1, #1

    ldr r0, .L133+12

    bl  fwrite

    ldr r3, [fp, #8]

    mvn r2, #0

    str r2, [r3]

.L128:

    ldr r3, [fp, #12]

    ldr r3, [r3]

    cmp r3, #1

    bne .L129

    ldr r3, [fp, #-8]

    mov r2, #17

    mov r1, #1

    ldr r0, .L133+16

    bl  fwrite

    ldr r3, [fp, #12]

    mvn r2, #0

    str r2, [r3]

.L129:

    ldr r3, [fp, #16]

    ldr r3, [r3]

    cmp r3, #1

    bne .L130

    ldr r3, [fp, #-8]

    mov r2, #14

    mov r1, #1

    ldr r0, .L133+20

    bl  fwrite

    ldr r3, [fp, #16]

    mvn r2, #0

    str r2, [r3]

.L130:

    ldr r3, [fp, #20]

    ldr r3, [r3]

    cmp r3, #1

    bne .L131

    ldr r3, [fp, #-8]

    mov r2, #16

    mov r1, #1

    ldr r0, .L133+24

    bl  fwrite

    ldr r3, [fp, #20]

    mvn r2, #0

    str r2, [r3]

.L131:

    ldr r3, [fp, #24]

    ldr r3, [r3]

    cmp r3, #1

    bne .L132

    ldr r3, [fp, #-8]

    mov r2, #16

    mov r1, #1

    ldr r0, .L133+28

    bl  fwrite

    ldr r3, [fp, #24]

    mvn r2, #0

    str r2, [r3]

.L132:

    ldr r1, [fp, #-8]

    mov r0, #10

    bl  fputc

    nop

    sub sp, fp, #4

    @ sp needed

    pop {fp, pc}

.L134:

    .align  2

.L133:

    .word   .LC41

    .word   .LC42

    .word   .LC43

    .word   .LC44

    .word   .LC45

    .word   .LC46

    .word   .LC47

    .word   .LC48

    .size   output\_move, .-output\_move

    .section    .rodata

    .align  2

.LC49:

    .ascii  "\012Player%d Wins!!\012\000"

    .align  2

.LC50:

    .ascii  "\012Player%d Losses... \012\000"

    .align  2

.LC51:

    .ascii  "\012\*\*\*Player%d Stats\*\*\*\012Hits: %d\012Misses: %d\012"

    .ascii  "Total Shots: %d\012Accuracy: %.2lf%%\012\012\000"

    .text

    .align  2

    .global output\_stats

    .syntax unified

    .arm

    .fpu vfp

    .type   output\_stats, %function

output\_stats:

    @ args = 32, pretend = 8, frame = 8

    @ frame\_needed = 1, uses\_anonymous\_args = 0

    sub sp, sp, #8

    push    {fp, lr}

    add fp, sp, #4

    sub sp, sp, #24

    str r0, [fp, #-8]

    add r1, fp, #4

    stm r1, {r2, r3}

    ldr r3, [fp, #28]

    cmp r3, #1

    bne .L136

    ldr r3, [fp, #4]

    mov r2, r3

    ldr r1, .L138

    ldr r0, [fp, #-8]

    bl  fprintf

.L136:

    ldr r3, [fp, #28]

    cmp r3, #0

    bne .L137

    ldr r3, [fp, #4]

    mov r2, r3

    ldr r1, .L138+4

    ldr r0, [fp, #-8]

    bl  fprintf

.L137:

    ldr ip, [fp, #4]

    ldr lr, [fp, #8]

    ldr r1, [fp, #12]

    ldr r0, [fp, #16]

    ldrd    r2, [fp, #20]

    strd    r2, [sp, #8]

    str r0, [sp, #4]

    str r1, [sp]

    mov r3, lr

    mov r2, ip

    ldr r1, .L138+8

    ldr r0, [fp, #-8]

    bl  fprintf

    nop

    sub sp, fp, #4

    @ sp needed

    pop {fp, lr}

    add sp, sp, #8

    bx  lr

.L139:

    .align  2

.L138:

    .word   .LC49

    .word   .LC50

    .word   .LC51

    .size   output\_stats, .-output\_stats

    .ident  "GCC: (Raspbian 6.3.0-18+rpi1) 6.3.0 20170516"

    .section    .note.GNU-stack,"",%progbits