

# Week 10 Weekly Test Sample Answers

## Test Conditions

These questions must be completed under self-administered exam-like conditions. You must time the test yourself and ensure you comply with the conditions below.

- You may complete this test in CSE labs or elsewhere using your own machine.
- You may complete this test at any time before **Wed Nov 25 21:00:00 2020**.
- Weekly tests are designed to act like a past paper - to give you an idea of how well you are progressing in the course, and what you need to work on. Many of the questions in weekly tests are from past final exams.
- Once the first hour has finished, you must submit all questions you've worked on.
- You should then take note of how far you got, which parts you didn't understand.
- You may choose then to keep working and submit test question anytime up to Wed Nov 25 21:00:00 2020
- However the maximum mark for any question you submit after the first hour will be 50%

You may access this **language documentation** while attempting this test:

- [C quick reference](#)
- [MIPS quick reference](#)

You may also access manual entries (the `man` command).

**Any violation of the test conditions will result in a mark of zero for the entire weekly test component.**

Set up for the test by creating a new directory called `test10`, changing to this directory, and fetching the provided code by running these commands:

```
$ mkdir test10
$ cd test10
$ 1521 fetch test10
```

Or, if you're not working on CSE, you can download the provided code as a [zip file](#) or a [tar file](#).

WEEKLY TEST QUESTION:

## Test if An Environment Variable contains A Non-empty String

Write a C program, `is_var_set.c`, which takes one argument, the name of environment variable

If the environment variable is set to a non-empty string, it should print **1**, otherwise it should print **0**.

```
$ gcc is_var_set.c -o is_var_set
$ ./is_var_set PATH
1
$ ./is_var_set PAT
0
$ VAR1=hello ./is_var_set VAR1
1
$ VAR2='' ./is_var_set VAR2
0
$ ./is_var_set VAR3
0
```

### NOTE:

The above example assumes there are not variables named `PAT` and `VAR3` already set your in environment. No error checking is required.

Your program can assume it is always a single argument.

It should always one line of output. The line of output should contain only 0 or 1.

Your solution must be in C only.

You are not permitted to run external programs. You are not permitted to use `system`, `popen`, `posix_spawn`, `fork` or `exec`.

When you think your program is working you can autotest to run some simple automated tests:

```
$ 1521 autotest is_var_set
```

When you are finished working on this exercise you must submit your work by running **give**:

```
$ give cs1521 test10_is_var_set is_var_set.c
```

Sample solution for is\_var\_set.c

```
// test the value of two environment are the same
// if so print 1, else print 0

#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

int main(int argc, char *argv[]) {
    assert(argc == 2);
    char *value = getenv(argv[1]);
    if (value != NULL && value[0] != '\0') {
        printf("1\n");
    } else {
        printf("0\n");
    }
    return 0;
}
```

WEEKLY TEST QUESTION:

## Test if A Pathname is a Directory

Write a C program, is\_directory.c, which takes one argument a pathname.

If the pathname exists and is a directory, it should print **1**, otherwise it should print **0**.

```
$ dcc is_directory.c -o is_directory
$ mkdir test_directory
$ echo >test_file
$ ./is_directory test_directory
1
$ ./is_directory test_file
0
$ ./is_directory non_existant_path
0
$ ./is_directory /home/cs1521/public_html
1
$ ./is_directory /etc/resolv.conf
0
```

### NOTE:

You can assume the pathname is not a symbolic link.

No error checking is required.

Your program can assume it is always a single argument.

It should always one line of output. The line of output should contain only 0 or 1.

Your solution must be in C only.

You are not permitted to run external programs. You are not permitted to use system, popen, posix\_spawn, fork or exec.

When you think your program is working you can autotest to run some simple automated tests:

```
$ 1521 autotest is_directory
```

When you are finished working on this exercise you must submit your work by running **give**:

```
$ give cs1521 test10_is_directory is_directory.c
```

Sample solution for is\_directory.c

```
// test the value of two environment are the same
// if so print 1, else print 0
```

```
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <assert.h>

int main(int argc, char *argv[]) {
    assert(argc == 2);
    struct stat s;
    if (
        stat(argv[1], &s) == 0 &&
        (s.st_mode & S_IFMT) == S_IFDIR
    ) {
        printf("1\n");
    } else {
        printf("0\n");
    }
    return 0;
}
```

WEEKLY TEST QUESTION:

## Test if Our Diary is Publically Readable

Write a C program, `is_diary_public.c`, which prints 1 if a file named `$HOME/.diary` exists and is publically readable.

It should print 0 otherwise.

By **\$HOME** we mean the value of the environment variable `HOME`.

```
$ dcc is_diary_public.c -o is_diary_public
$ echo >$HOME/.diary
$ chmod 644 $HOME/.diary
$ ./is_diary_public
1
$ chmod 600 $HOME/.diary
$ ./is_diary_public
0
$ rm $HOME/.diary
$ ./is_diary_public
0
```

### NOTE:

You only need to check if the file is readable by users who are not the owner of the file and don't have group access to the file. It is possible, but not normally useful, to set permissions so a file is readable by other users but not readable by the owner of the file. You do not need to consider this case.

You can assume if `$HOME/.diary` exists its is an ordinary file.

No error checking is required.

Your program can assume it is always a single argument.

It should always one line of output. The line of output should contain only 0 or 1.

Your solution must be in C only.

You are not permitted to run external programs. You are not permitted to use `system`, `popen`, `posix_spawn`, `fork` or `exec`.

When you think your program is working you can autotest to run some simple automated tests:

```
$ 1521 autotest is_diary_public
```

When you are finished working on this exercise you must submit your work by running **give**:

```
$ give cs1521 test10_is_diary_public is_diary_public.c
```

Sample solution for `is_diary_public.c`

```

#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]) {
    char *home_pathname = getenv("HOME");
    if (home_pathname == NULL) {
        home_pathname = ".";
    }

    char *basename = ".diary";
    int diary_pathname_len = strlen(home_pathname) + strlen(basename) + 2;
    char diary_pathname[diary_pathname_len];
    snprintf(diary_pathname, sizeof diary_pathname, "%s/%s", home_pathname, basename);

    struct stat s;
    if (
        stat(diary_pathname, &s) == 0 &&
        (s.st_mode & 004)
    ) {
        printf("1\n");
    } else {
        printf("0\n");
    }

    return 0;
}

```

## Submission

When you are finished each exercise make sure you submit your work by running **give**.

You can run **give** multiple times. Only your last submission will be marked.

Don't submit any exercises you haven't attempted.

If you are working at home, you may find it more convenient to upload your work via [give's web interface](#).

Remember you have until **Wed Nov 25 21:00:00 2020** to complete this test.

Automarking will be run by the lecturer several days after the submission deadline for the test, using test cases that you haven't seen: different to the test cases `autotest` runs for you.

(Hint: do your own testing as well as running `autotest`)

## Test Marks

After automarking is run by the lecturer you can [view it here](#) the resulting mark will also be available via [via give's web interface](#) or by running this command on a CSE machine:

```
$ 1521 classrun -sturec
```

The test exercises for each week are worth in total 1 marks.

The best 6 of your 8 test marks for weeks 3-10 will be summed to give you a mark out of 9.

COMP1521 20T3: Computer Systems Fundamentals is brought to you by  
the [School of Computer Science and Engineering](#)  
at the [University of New South Wales](#), Sydney.

For all enquiries, please email the class account at [cs1521@cse.unsw.edu.au](mailto:cs1521@cse.unsw.edu.au)

CRICOS Provider 00098G