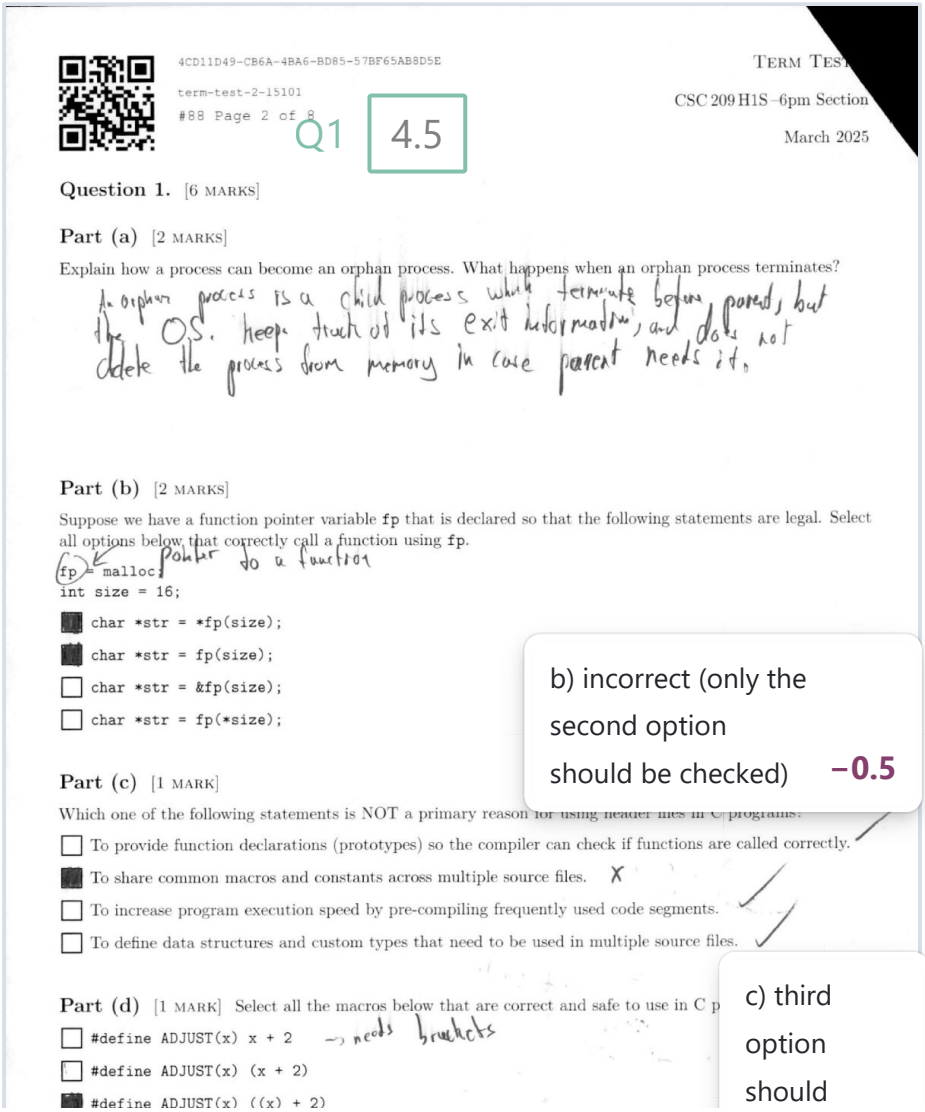
Term Test 2 - L5101

My score

87.5% (22.75/26)



4CD11D49-CB6A-4BA6-BD85-57BF65AB8D5E
term-test-2-15101
#88 Page 2 of 8

Q1 4.5

TERM TEST
CSC 209 H1S –6pm Section
March 2025

Question 1. [6 MARKS]

Part (a) [2 MARKS]

Explain how a process can become an orphan process. What happens when an orphan process terminates?

An orphan process is a child process which terminate before parent, but the OS. keeps track of its exit information, and does not delete the process from memory in case parent needs it.

Part (b) [2 MARKS]

Suppose we have a function pointer variable `fp` that is declared so that the following statements are legal. Select all options below that correctly call a function using `fp`.

`fp = malloc(` Pointer to a function
`int size = 16;`

- [x] `char *str = *fp(size);`
- [x] `char *str = fp(size);`
- [ ] `char *str = &fp(size);`
- [ ] `char *str = fp(*size);`

b) incorrect (only the second option should be checked) −0.5

Part (c) [1 MARK]

Which one of the following statements is NOT a primary reason for using header files in C programs?

- [ ] To provide function declarations (prototypes) so the compiler can check if functions are called correctly.
- [x] To share common macros and constants across multiple source files. X
- [ ] To increase program execution speed by pre-compiling frequently used code segments.
- [ ] To define data structures and custom types that need to be used in multiple source files.

Part (d) [1 MARK] Select all the macros below that are correct and safe to use in C p

- [ ] `#define ADJUST(x) x + 2` → needs brackets
- [ ] `#define ADJUST(x) (x + 2)`
- [x] `#define ADJUST(x) ((x) + 2)`

c) third option should

☐ #define ADJUST(x) (x) + 2

be checked −1

RM TEST 2

CSC 209 H1S –6pm Section

March 2025

Q2 2.25

FD14DC31-ABBF-48AA-9FA5-D64929063CC3

term-test-2-15101

#88 Page 3 of 8

Question 2. [3 MARKS]

Part (a) [2 MARKS]

Assume the following program runs without error. The `echo` program simply prints its arguments to `stdout`.

```
int main() {
    char p[] = "A";

    if (fork() == 0) {
        p[0] = 'B';
    } else {
        p[0] = 'C';
    }
    execlp("echo", "echo", p, NULL);
    printf("D\n");

    return 0;
}
```

What will be printed to the screen when this program is run? Check all that apply.

☐ A
■ B
■ C
■ D

a) incorrect −0.5

Part (b) [1 MARK]

The complete contents of `code.c` are shown below. Which of the following compilation commands will succeed without any warning or errors? Check all that apply.

```
int f(int);

int g() {
    return f(10);
}
```

☐ gcc -Wall -g -o code code.c
☐ gcc -Wall -g -c code.c
☐ gcc -g code.c
■ gcc -c code.c

b) incorrect −0.25

Q3ab 2

Question 3. [5 MARKS] The current working directory contains only the following files:

`Makefile` `config.h` `main.c` `math_ops.c` `math_ops.h` `matrix_ops.c` `matrix_ops.h`

The Makefile has the following contents:

```
1  all: calc
2
3  calc: main.o math_ops.o matrix_ops.o
4          gcc -Wall -g -o calc main.o math_ops.o matrix_ops.o -lm
5
6  main.o: main.c math_ops.h matrix_ops.h config.h
7          gcc -Wall -g -c main.c
8
9  math_ops.o: math_ops.c math_ops.h config.h
10         gcc -Wall -g -c math_ops.c
11
12 matrix_ops.o: matrix_ops.c matrix_ops.h math_ops.h
13         gcc -Wall -g -c matrix_ops.c
14
15
16 clean:
17         rm -f *.o calc
```

Part (a) [1 MARK] If you **first** run `make matrix_ops.o` which files are created or modified?

The only file created is matrix_ops.o object file.

Part (b) [1 MARK] If you **next** run `make calc` which actions are executed? (This command is run after the command in part (a).)

actions created: checks main.o dependency, runs
gcc -Wall -g -c main.c
also checks math_ops.o
so: gcc -Wall -g -c math_ops.c
also lastly gcc -Wall -g -o calc main.o math_ops.o matrix_ops.o -lm

all correct 2

Q3cd 3

Part (c) [1 MARK] Which rule is evaluated first when you run `make`?

all: calc

Part (d) [2 MARKS] Suppose we replace the rules `main.o`, `math_ops.o`, `matrix_ops.o` with the following rule. Does this rule work the same as the three rules it replaces? Explain your answer.

```
%.o: %.c
    gcc -Wall -g -c $<
```

This rule usually works with % wildcard matching object files and header files with the same file name. But in this case, object files depend on more than 21 different header files and so it will not work as expected.

all correct 3

Q4 6

C4AE157A-038A-44FE-9A7E-41F55B8F1A37

#88 Page 6 of 8

TERM TEST

CSC 209 H1S –6pm Section

March 2025

Question 4. [6 MARKS] Consider the following program that runs to completion without error.

```
1  int main() {
2
3      if(fork() > 0) {
4          if(fork() == 0) {
5              fprintf(stderr, "A");
6              exit(0);
7          } else {
8              fprintf(stderr, "B");
9              exit(0);
10         }
11
12     } else {
13         if(fork() == 0) {
14             fprintf(stderr, "C");
15         } else {
16             wait(NULL);
17             fprintf(stderr, "E");
18         }
19     }
20     fprintf(stderr, "D");
21     return 0;
22 }
```

Parent

P1 fork()>0 P2 Processes: 2 3

Stderr waits for grandchild

P3 P4

can be printed more than once

Part (a) [1 MARK] Including the process that runs `main`, how many processes are created when this program is run?

parent

4

Part (b) [1 MARK] Which letter or letters cannot be printed first?

D, E,

Part (c) [4 MARKS] For each statement below, indicate whether the statement is true or false by circling the correct answer.

TRUE	FALSE	All of the characters must be printed before the shell prompt appears.
TRUE	FALSE	At least one character is printed more than once.
TRUE	FALSE	B can be printted last.
TRUE	FALSE	D cannot be printed before C.

All correct **6**

Q5 5

RM TEST 2

CSC 209 H1S –6pm Section

March 2025

6F94CD9E-A1A1-4798-A741-8C975668A2A2

term-test-2-15101

#88 Page 7 of 8

Question 5. [6 MARKS]

Complete the following function that operates on files containing binary data of the format described in the comments. You do **not** need to do any error checking.

```
struct data { /* definition omitted */ };

/* replace_data replaces the kth element of the data array in the file named filename with
   new_data. If k is not within the bounds of the array, return -1 and leave the file
   unchanged. Otherwise update the file and return 0. Assume the data array is 0-indexed.

   The format of the binary file is as follows:
   - The file first contains the length of the data array.
   - The file then contains the data array.

   For example, for a file "foo.dat" containing:
   - the integer 10 (4 bytes)
   - the data array (10 * sizeof(struct data) bytes)
   A call to replace_data("foo.dat", 0, new_data) would update "foo.dat" to:
   - the integer 10 (4 bytes)
   - the contents of new_data (sizeof(struct data) bytes)
   - the original last 9 elements of the data array (9 * sizeof(struct data) bytes)

   Your solution should minimize the number of read and write calls.*/

int replace_data(char *filename, int k, struct data new_data) {
    FILE *file_name= fopen( filename, "r+" );
    int size = freads ( &size, sizeof(int), 1, file_name );
    fseek(filen-name, k*sizeof(struct data), SEEK_CUR);
    if (k>size) { return -1; }
    if (fork()==0){
        FILE *write_file = fop
        fseek(write_file, k* size of(struct data)+4, SEEK_CUR);
        fwrite( &new_data, sizeof(new_data), 1, write_file);
        fclose( write_file );
        exit(0); }
    fclose(file_name); return 0;
}
```

Incorrect bound check **–1**

"Blank" page for rough work or answers.

Total Marks = 26