

Student Name: _____

Student ID: _____

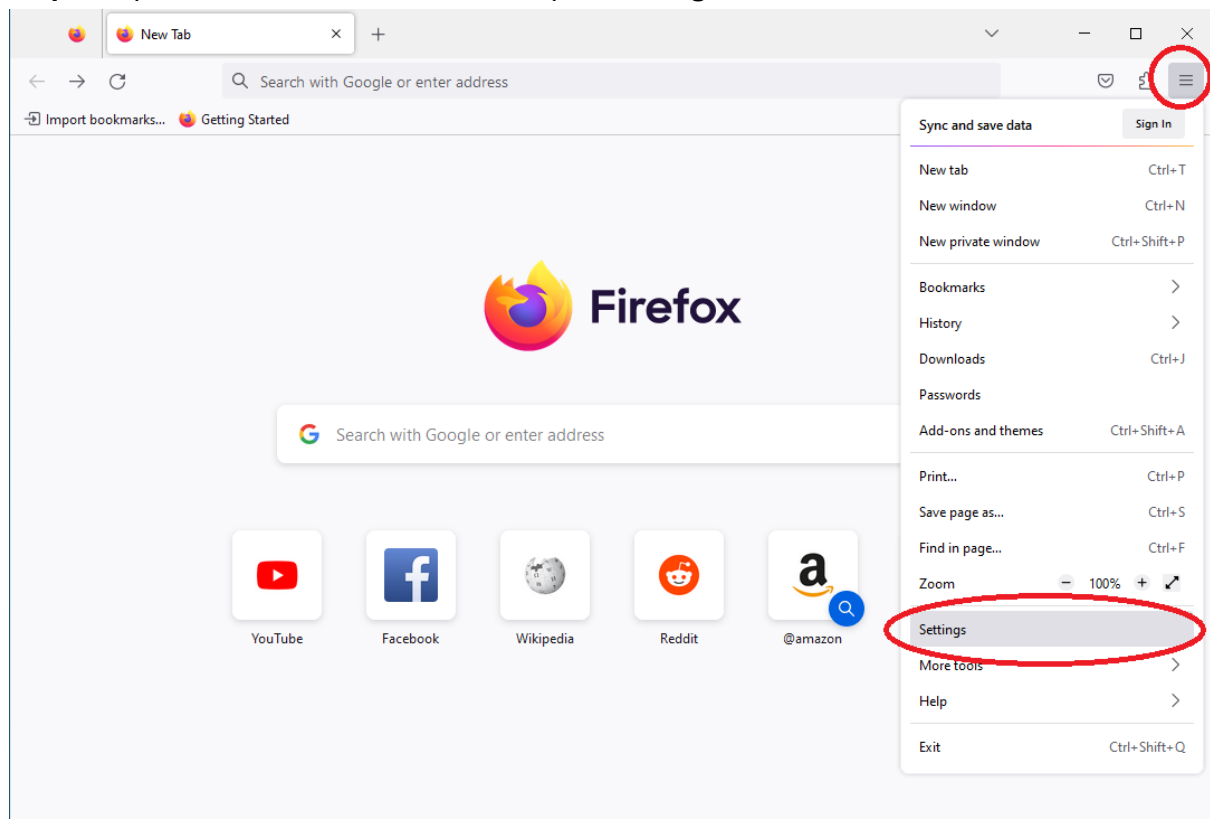
Rules:

1. Time limit: 90 minutes
2. Unless otherwise specified, use C to answer the programming problems in this test.
3. Answer all questions. Write your answers in the answer book.
4. Write your name and SID in both the question paper and the answer book.
5. Except the PC provided, you are not allowed to use any other electronic devices during the test.
6. You have restricted Internet access for browsing some useful webs in the whitelist after setting up the proxy as instructed.

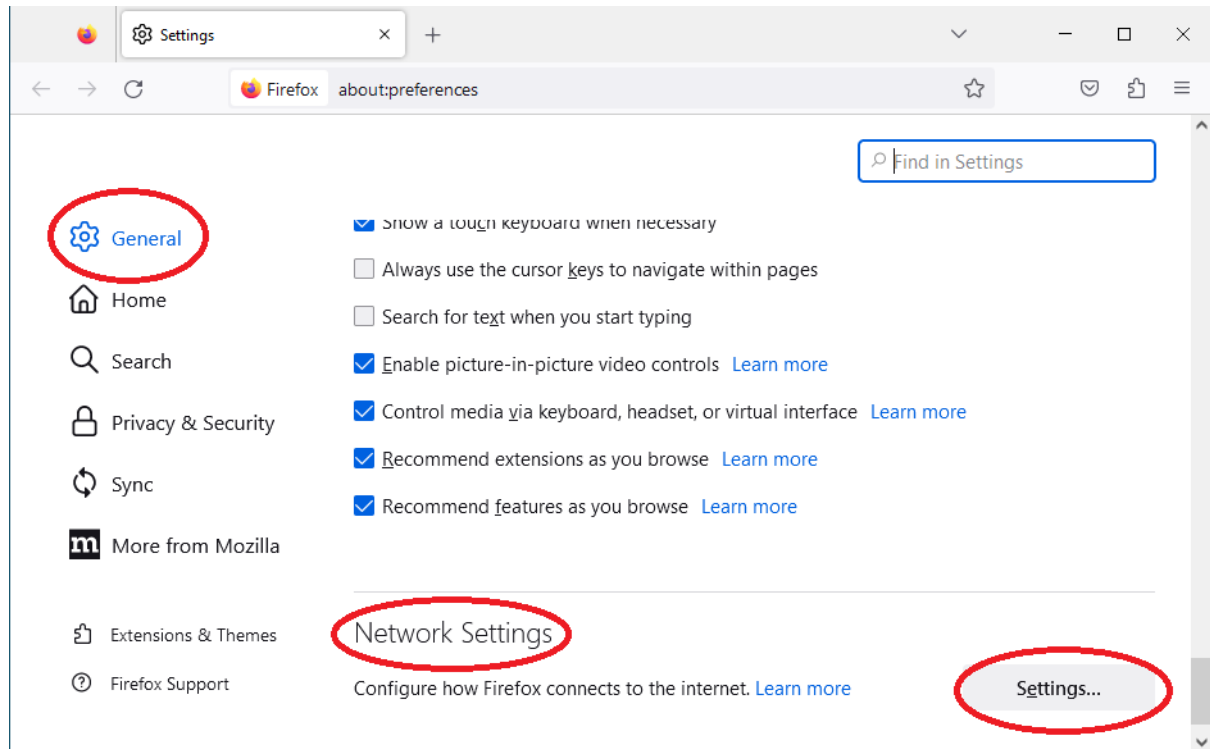
Proxy Configuration

You may set up the proxy following the instructions below for accessing the (restricted) Internet.

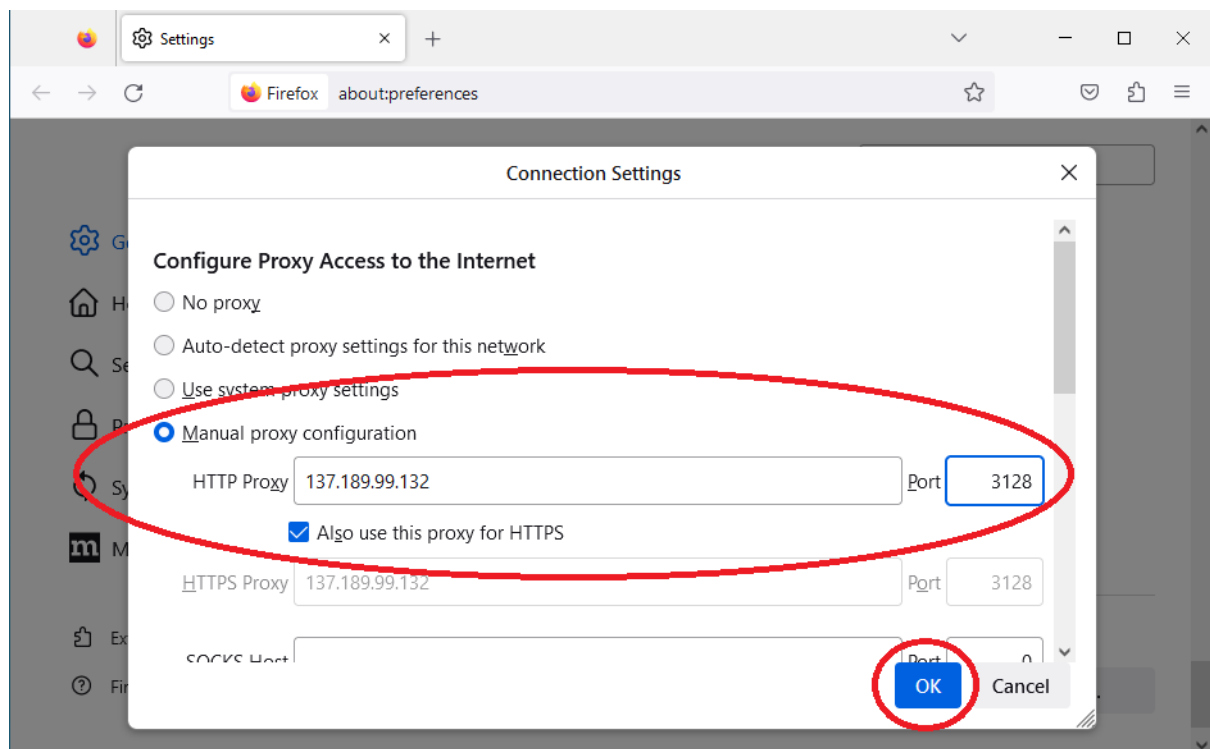
Step 1: Open the web browser **Firefox**. Open **Settings**.



Step 2: Go to General → Network Settings → Settings...



Step 3: Select Manual proxy configuration. Tick Also use this proxy for HTTPS. Enter 137.189.99.132 in the HTTP Proxy textbox, and enter 3128 in the Port textbox. Then, press OK.



Step 4: Try accessing the Blackboard using Firefox and see if your configuration is fine.
<https://blackboard.cuhk.edu.hk/>

Some useful webs in the whitelist:

- Lecture notes
<https://blackboard.cuhk.edu.hk/>
- Man pages
<https://man7.org/linux/man-pages/>
- C standard library reference
https://www.tutorialspoint.com/c_standard_library/

Note on Visual Studio

For those who want to use Visual Studio, you need to put the following

```
#define _CRT_SECURE_NO_WARNINGS
```

at the beginning of your code, i.e., even before `#include<stdio.h>`

Otherwise, Visual Studio identifies some functions like `scanf()` as “unsafe” and gives compilation errors.

For example:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int i;
    scanf("%d", &i);
    return 0;
}
```

Part I: Multiple Choices (25%)

Each question in this part takes 5%.

1. Which of the following file names is not valid in Linux? (Assume using a standard file system like ext4)

- A. !!!
- B. 2023Feb
- C. 02/2023**
- D. It is Feb now

2. Which of the following is the absolute mode of the permission -w-r-xrw- ?

- A. 652
- B. 256**
- C. 642
- D. 246

3. Suppose user K wants to upload the file `top_secret.txt` from his PC to his home directory in the remote host `mib` using `scp`. Which of the following actions can achieve this task?

- A. Run `scp top_secret.txt K@mib:~` on the remote host.
- B. Run `scp top_secret.txt K@mib:~` on his PC.**
- C. Run `scp K@mib:~/top_secret.txt ./` on the remote host.
- D. Run `scp K@mib:~/top_secret.txt ./` on his PC.

4. The status of a process can be shown using `top`. Suppose `top` shows the following:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
101	john	20	0	1024	2080	1110	Z	0.0	0.0	0:01.00	wwz

The status of the process shown above is Z. What does Z stand for?

- A. Running
- B. Sleeping
- C. Stopped
- D. Zombie**

5. Suppose the following is the output of `df -h`

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sdb	251G	11G	229G	5%	/
tmpfs	13G	0	13G	0%	/mnt/wsl
tools	238G	211G	28G	89%	/init
none	13G	8.0K	13G	1%	/run
none	13G	0	13G	0%	/run/lock
none	13G	0	13G	0%	/run/shm
none	13G	0	13G	0%	/run/user
tmpfs	13G	0	13G	0%	/sys/fs/cgroup
drivers	238G	211G	28G	89%	/usr/lib/wsl/drivers
lib	238G	211G	28G	89%	/usr/lib/wsl/lib
C:\	238G	211G	28G	89%	/mnt/c
E:\	466G	42G	425G	9%	/mnt/e

Which of the following commands can be used to extract the Use% column of C:\ and E:\ ?
That is, the output is

89%
9%

- A. `df -h | tr -s ' ' | cut -d ' ' -f 5 | grep -e "[\\\"]"`
- B. `df -h | cut -d ' ' -f 5 | tr -s ' ' | grep -e "[\\\"]"`
- C. `df -h | grep -e "[\\\"]" | tr -s ' ' | cut -d ' ' -f 5`**
- D. `df -h | grep -e "[\\\"]" | cut -d ' ' -f 5 | tr -s ' '`

Part II: Short Questions (35%)

6. Write the Vim command that can be used to replace all words apple into orange in the current line. (4%)

`:s/\<apple\>/orange/g`

g 1 mark (accept gc, ONLY accept if it is put after the last /)

\<apple\> 1 mark (accept wrong order of \<apple\> and orange)

orange 1 mark (accept wrong order of \<apple\> and orange)

s/ / 1 mark for formatting

(no mark for %s or other type of s before the first /)

(**NO** mark for wrong order of \<apple\> and orange)

Accept no : before s

7. Write a line of Linux command to print the fifth line of the file `log.txt` in stdout. (4%)

`head -5 log.txt | tail -1`

head -5 1 mark (accept -n 5 or -n5)
 log.txt 1 mark
 (ONLY accept as a parameter of the first command)
 (i.e., accept wrong command but correct filename)
 tail -1 1 mark (accept -n 1 or -n1)
 | 1 mark (ONLY accept between commands)

Accept cat the file and then pipe to head
 Accept using redirected input for head

8. Write a line of Linux command that greps only the lines that start with meow and contain the word cat in the file animal.txt (4%)

Example: Suppose the following is the content of animal.txt

```
meowing dancing cat is awesome
meowing jumping dog is nice
barking climbing cat is great
barking sleeping dog is cool
```

Then, your command should output

```
meowing dancing cat is awesome
```

Note that the above sample is an example only. There can be more than one line in the output.

grep -e "^meow" animal.txt | grep -e "\<cat\>"

Accept reversing the order of grep, i.e.,
grep -e "\<cat\>" animal.txt | grep -e "^meow"

animal.txt 0.5 mark (ONLY accept as a parameter of the first command)
 meow 0.5 mark (accept \<meow\>)
 ^ 0.5 mark (ONLY accept putting as the first character in the
 regexp)
 cat 0.5 mark
 \< \> 0.5 mark
 grep 0.5 mark (accept only one grep)
 -e 0.5 mark (accept -E)
 | 0.5 mark

Accept using single quote
 Accept using -E instead of -e

Accept cat the file and then pipe to grep
Accept using redirected input for grep

Alternative solution:

grep -e "^meow.*\<cat\>.*" animal.txt

Accept putting \$ at the end of regexp

animal.txt 0.5 mark
meow 0.5 mark (accept \<meow\>)
^ 0.5 mark (ONLY accept putting as the first character in the
regexp)
cat 0.5 mark
\< \> 0.5 mark
grep 0.5 mark (accept only one grep)
-e 0.5 mark (accept -E)
.* 0.5 mark (for both .*) (NO mark for \.)

Accept using single quote
Accept using -E instead of -e
Accept cat the file and then pipe to grep
Accept using redirected input for grep

9. The following is a list of users in a Linux machine somewhere in the world.

User Name	Group Name
bob	student
alice	student
danny	staff
hoover	staff
root	root

The following is the output of `ls -l` in one of the directories.

```
total 12
-rw----r-- 1 danny  staff  0 Feb 16 10:30 a.txt
-rw-r----- 1 alice  staff  0 Feb 16 10:30 b.txt
-r--rw-r-- 1 hoover student 0 Feb 16 10:30 c.txt
-rw-rw---- 1 bob    student 5 Feb 16 10:30 d.txt
```

Start from full marks. Minus 1 for each wrong answer (or missing answer). NO mark for no answer.

a) Which user(s) cannot read a.txt ? (2%)

hoover

b) Which user(s) cannot read b.txt ? (2%)

bob

c) Which user(s) cannot write c.txt ? (2%)

hoover, danny

d) Which user(s) cannot write d.txt ? (2%)

hoover, danny

e) Which user(s) can execute d.txt? (2%)

No one (or bob, alice, root)

Take the condition with higher marks

10. Peter has 4 files in a directory on his PC. The following is the output of `ls -l` in this directory.

```
total 12
-rw-r--r-- 2 peter peter 0 Feb 16 10:30 a.txt
-rw-r--r-- 2 peter peter 0 Feb 16 10:30 b.txt
-rw-r--r-- 1 peter peter 0 Feb 16 10:30 c.txt
lrwxrwxrwx 1 peter peter 5 Feb 16 10:30 d.txt -> c.txt
```

Peter runs the commands

```
echo hello > a.txt
```

```
echo bye > d.txt
```

After that, the output of `ls -l` becomes

```
total 12
-rw-r--r-- 2 peter peter 6 Feb 16 10:30 a.txt
-rw-r--r-- 2 peter peter 6 Feb 16 10:30 b.txt
-rw-r--r-- 1 peter peter 4 Feb 16 10:30 c.txt
lrwxrwxrwx 1 peter peter 5 Feb 16 10:30 d.txt -> c.txt
```

a) There are only 5 characters in the string `hello`. Why does the file size of `a.txt` change to 6 bytes? (2%)

Any valid answers: (full marks if there is the keyword `newline` or `'\n'`)

There is a newline (`'\n'`) character at the end of the file.

echo produces a newline character at the end of the output.

etc.

Give 1 mark if only answering `hello` has 5 bytes, or there is a null character at the end (although there is not)

b) Why does the file size of `c.txt` change? (2%)

Full marks if there is the keyword `symbolic link` or `shortcut`.

Give 1 mark if answering the two files are referring to the same file.

Give NO mark if answering the two files are the same file.

c) Why is the file size of d.txt 5 bytes? (2%)

- Hints: The answer is actually shown in the output of `ls -l`

Any expression about :

The length (number of characters) of the target filename

d) The content of b.txt is also changed to hello. Suggest a reason. (2%)

Any keyword about hard link

Give 1 mark if saying that they are the same file (or inode) / referring to the same file.

11. Alan wants to test the behavior of his program when it is crashed by segmentation fault. He named his program `be_fun` before he executed it. The following is the output of `ps`

PID	TTY	TIME	CMD
12	pts/0	00:00:00	bash
18	pts/0	00:00:00	seg_fault
24	pts/0	00:01:23	be_fun
30	pts/0	00:03:21	be_sad
36	pts/0	00:00:00	ps

a) What command should Alan type in the terminal to forcefully make the process he wants to test segmentation fault? (3%)

kill -11 24

kill 1 mark

24 1 mark

-11 or -s 11 or -s SIGSEGV 1 mark

Alternative answer

killall -11 be_fun

killall 1 mark (accept using pkill)

be_fun 1 mark

-11 or -s 11 or -s SIGSEGV 1 mark

b) The default behavior for segmentation fault is to terminate the process. However, after Alan forcefully makes his process segmentation fault, he finds that the process is not terminated. This is as he expected because he did "something" in the code of his program. Suggest what has Alan done to his code? (2%)

The signal is handled / ignored.

Give 1 mark for, the process is stopped or (uninterruptible) sleep.

Part III: C Programming (40%)

Each of the following problems takes 20%.

12. Given a 11×11 char array that represents a maze.

- The boundaries of the maze, i.e., `maze[0][i]`, `maze[i][0]`, `maze[10][i]` and `maze[i][10]` for all $i = 0, 1, \dots, 10$, are walls.
- The wall is represented by the character `#`, and the passage is represented by spaces.
- The starting point of the maze is at `maze[1][1]`, and the exit of the maze is at `maze[9][9]`.
- In every step, you can move to the passage either on the top, bottom, left, or right only.
- You cannot pass through walls or break walls.

Task: Write a function with the following prototype

```
int can_exit(char maze[11][11], int x, int y);
```

that returns 1 if there is a path from the “starting point” to the exit, or returns 0 if there is not.

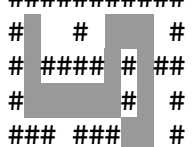
The purpose of the parameters are as follows:

- `char maze[11][11]` The maze itself
- `int x, int y` The “starting point” is at `maze[y][x]`

That is, the output of `can_exit(maze, 1, 1)`; is the answer we desired.

Here are the guidelines:

- You are allowed to modify the content of the array `maze`.
- Assume the array `maze` is initialized at the beginning.
- You do not need to write down the function `main()` in the answer book.
- You can assume the necessary header files are included.

Sample maze 1	Sample maze 2
<pre>##### # # # # ##### # ## # # # ### ### # # ## ### # ### ### # # # ### # # ### # # # # # # #####</pre>	<pre>##### # # # # ##### ## # # # ##### # # ## ### # ### ### # # # ### # # ### # # # # # # #####</pre>
Sample return value 1	Sample return value 2
1	0
Reason	Reason
<pre>##### # # # # ##### # ## # # # ### ### #</pre> 	The starting point and the exit are isolated.

<pre># ## ### # ### ### # # # ### # # ### # # # # # # #####</pre>	
--	--

Hints:

- Fill the visited passages into walls, so that you won't move into visited passages again during the search.
- Use recursion to perform new searches starting at nearby passages.

For each up, bottom, left, right direction: (omit syntax issue at this point)

Check if the direction a wall/passage 1 mark

Call recursion on this direction 1 mark

Correct parameters for the recursion 1 mark

Correct logic for this direction 1 mark

Fill the current location (or the wall at the next step) into # 1 mark

Detect reaching the exit 1 mark

Correct return flow for the answer 1 mark

Correct Syntax 1 mark

13. We want to find the minimal number of steps we need to modify a string `str1` into another string `str2`. In each step, we can perform one of the following operations:

- Insert one character at an arbitrary position of the string.
- Delete one character at an arbitrary position of the string.
- Substitute one character at an arbitrary position of the string into another character.

As the above operations are symmetric, you can perform them on either `str1` or `str2`.

Here are the assumptions of this question:

- The strings `str1` and `str2` are null-terminated.
- The strings `str1` and `str2` have lengths at most 10 (excluding the null character `\0`).

Task: Write a function with the following prototype

```
int steps(char *str1, char *str2);
```

that returns the minimal number of steps to modify `str1` into `str2`.

Here are the guidelines:

- You do not need to write down the function `main()` in the answer book.
- You can assume the necessary header files are included.

Example 1		Example 2	
<code>str1</code>	<code>kitten</code>	<code>str1</code>	<code>trolley</code>

str2	sitting	str2	tolling
Answer 1		Answer 2	
3		4	
Reason		Reason	
kitten → sitten (Substitution) sitten → sittin (Substitution) sittin → sitting (Insertion)		trolley → tolley (Deletion) tolley → tolliy (Substitution) tolliy → tollin (Substitution) tollin → tolling (Insertion)	

Hints:

- Use `strlen()` to obtain the lengths of the strings.
- Let `str1` be $a_1a_2 \dots a_m$ and `str2` be $b_1b_2 \dots b_n$, where each a_k and b_k is a character. Let $d(i, j)$ be the minimal number of steps to modify the substring $a_1a_2 \dots a_i$ into the substring $b_1b_2 \dots b_j$.
 - Express $d(i, j)$ in terms of $d(i, j - 1)$, $d(i - 1, j)$ and $d(i - 1, j - 1)$ under different cases.
 - If $a_i \neq b_j$, perform one of the 3 operations that results in the minimal number of steps.
 - Consider the boundary cases $d(i, 0)$ and $d(0, j)$ separately.
- Use a 2D array for storing the computed values of $d(i, j)$ to avoid recursion.

Use `strlen()` to get the length of the input strings 1 mark

Consider the case when $a_i = b_j$ 1 mark

Use the value of $d(i-1, j-1)$ (maybe wrong usage) 1 mark

Set $d(i, j)$ equals $d(i-1, j-1)$ 1 mark

Boundary conditions ($i, j = 0$ or -1) 1 mark

Consider the case when $a_i \neq b_j$ 1 mark

Use the value of $d(i, j-1)$, $d(i-1, j)$ and $d(i-1, j-1)$ Each 1 mark

Boundary conditions ($i, j = 0$ or -1) (3 cases) Each 1 mark

Take the minimum of the above values 1 mark

+1 for the current answer (maybe wrong usage) 1 mark

Loop (or recursion) all characters in the string (accept $[0,10]$ of the string array)

Each 1 mark

Correct order of loop (or recursion)

Each dimension 1 mark

Correct logic

1 mark

Syntax

1 mark