**W1 -** HOMEWORK

*Array Problem Solving*

## *At the end of this practice, you should be able to…*

* Access to **previous, next elements** in an array
* Be able to check if **everything is** XX or at **least something is** XX in a list of numbers
* Refactor a LOOP for **avoid breaks statements**
* Use **debugging** **techniques** to check for errors

## *How do we structure exercises?*

We organize each practice into 3 parts:

| ANALYSE | **Understand** existing codes, find the **bugs** or **complete** missing gaps |
| --- | --- |
| MANIPULATE | Ensure you can **apply the theory** with some basic challenges |
| CREATE | **Express your creativity** with more complex challenges |

## *Are you lost?*

You can read the following documentation to be ready for this practice

<https://www.w3schools.com/c/c_arrays.php>

**EXERCISE 1** *COUNT THE NUMBER OF EVEN NUMBERS*

We want to **count the number of even numbers** in 1 array of 5 elements.

| INPUT | Numbers (int []) |
| --- | --- |
| OUTPUT | The number of even numbers |

Example

| INPUT | OUTPUT |
| --- | --- |
| {1, 2, 3, 4, 5} | 2 |
| {2, 4, 6, 8, 10} | 5 |
| {3, 3, 3, 3, 3} | 0 |

**Q1 –** How many variables do you need to solve this problem? *(You can add more rows…)*

| VARIABLE | TYPE | GOAL |
| --- | --- | --- |
| i | int | for loop |
| found | int | for counting the amount of even numbers |
| arr | int | for user input |

**Q2 –** How do you plan to **loop** on the array elements? Complete the table

| I will **start** at index | i =0 |
| --- | --- |
| I will **increment** of | i++ |
| I will **stop** at index | i<5 |

**Q3 –** Write first the pseudo code.

* Test it using the following example: [1,4,6,8,9]

set arr[5];

set found to 0;

print (Enter your array)

for (set i to 0 from 0 to 4)

user input

for (set i to 0 from 0 to 4)

if (arr[i] % 2 is equal to 0)

found = found plus 1

print(found)

**Q4 –** Once you are **clear with your pseudo code**, code your algorithm in C.

#include <stdio.h>

int main(){

int arr[5], found = 0;

printf("Enter your array :\n");

for(int i = 0; i < 5; i++){

scanf("%d", &arr[i]);

}

for (int i = 0; i < 5; i++){

if(arr[i] % 2 == 0){

found += 1;

}

}

printf("The amount of even numbers are : %d\n", found);

return 0;

}

**EXERCISE 2**  *MOVE ALL ZEROES TO THE END*

We have as input an array of 10 numbers.

We want to move all zeroes to the end, while maintaining the order of the non-zero elements.

*Note: You must use the same array (you cannot create a new array)*

| **INPUT** | int [10] | The array of 10 numbers |
| --- | --- | --- |
| **OUTPUT** | int [10] | The same array  *(All zeroes moved to the end)* |

Examples

| **INPUT** | **OUTPUT** |
| --- | --- |
| {1, 2, 3, 0, 4, 5, 0, 6, 7, 8} | {1, 2, 3, 4, 5, 6, 7, 8, 0, 0} |
| {0, 0, 0, 0, 0, 1, 1, 1, 1, 1} | {1, 1, 1, 1, 1, 0, 0, 0, 0, 0} |
| {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} | {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} |
| {0, 0, 0, 0, 0, 0, 0, 0, 0, 1} | {1, 0, 0, 0, 0, 0, 0, 0, 0, 0} |

**#include <stdio.h>**

**int main(){**

**int arr[10], num;**

**printf("Enter your array :\n");**

**for(int i = 0; i < 10; i++){**

**scanf("%d", &arr[i]);**

**}**

**for(int i = 0; i < 5; i++){**

**for (int j = 0; j < 9; j++ ){**

**if(arr[j] == 0){**

**num = arr[j];**

**arr[j] = arr[j + 1];**

**arr[j + 1] = num;**

**}**

**}**

**}**

**printf("\nsorted array : ");**

**for (int i = 0; i < 10; i++){**

**printf("%d,", arr[i]);**

**}**

**return 0;**

**}**

**EXERCISE 3**  *THE COVID25 CONTAMINATION*



A **research lab** in Phnom Penh is studying how a new virus (covid 25) spreads in the population.

People are **standing in a line** and each person can be:

* Healthy ('**H**')
* Infected (**'I**')
* Almost Recovered (**'A**')
* Recovered (**'R**')

The simulation will model how the infection spreads over time in the array of people.

We want to know, after 5 days, what is the contamination status of the population.

**Rules**

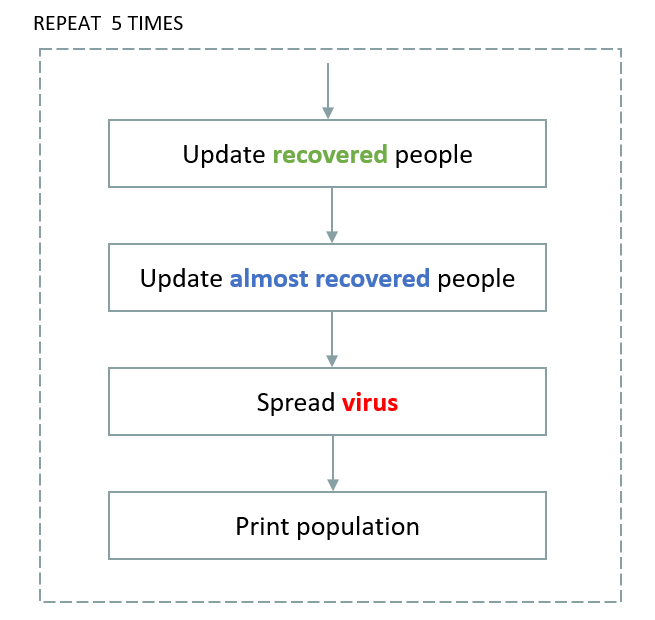
1. The simulation starts with **one infected** person in a population of **10 healthy people**.
2. Each day, an infected person **spreads the virus to its direct neighbors** (left / right)
3. After **1 day,** an infected person becomes **almost recovered.**
4. After **2 days,** an almost recovered person **recovers** and is **no longer contagious**!
5. The simulation runs for **5 days**.

Example

| DAY | CONTAMINATION STATUS | EXPLANATION |
| --- | --- | --- |
| DAY 1 | H H H H **I** H H H H H | *Index 4 infected* |
| DAY 2 | H H H **I** **A** **I** H H H H | *Index 3 and 5 infected* |
| DAY 3 | H H **I** **A** **R** **A** **I** H H H | *Index 4 recovered*  *Index 2 + 6 infected* |
| DAY 4 | H **I** **A** **R** **R** **R** **A** **I** H H | *Index 3 and 5 recovered*  *Index 1 + 7 infected* |
| DAY 5 | **I** **A** **R** **R** **R** **R** **R** **A** **I** H | *Index 2 and 6 recovered*  *Index 0 and 8 infected* |

**Program structure**

The bellow flow chart represents the 4 **operations** to be done during each day of your simulator:



Each operation will be implemented in a **separated function**, to easier the code understanding.

Therefore, our program will be organized as follows:

void updateRecovered() {

// TODO Alsmost infected people (A) becomes recovered (R)

}

void updateAlmostRecovered() {

// TODO Infected people (I) becomes alsmost infected (A)

}

void spreadVirus() {

// TODO Spread the virus, following the rules

}

void display() {

// TODO Display the population and the current day.

}

int main() {

display();

for (day = 1; day <= 5; day++) {

updateRecovered();

updateAlmostRecovered();

spreadVirus();

display();

}

printf("Simulation over!\n");

return 0;

}

We also provide, as a start code, the following constant (#define) and variables:

#define N 10 // Population size

#define D 5 // Number of days to simulate

char population[N]; // 'H', 'I', 'A', or 'R'

int day = 0; // Current day

**Your tasks**

**Q1 –** Implement the display() function to print the array and the day on console

**Q2 –** Implement the updateRecovered () function

**Q3 –** Implement the updateAlmostRecovered () function

**Q4 –** Implement the spreadVirus () function

**Q5 –** Test your code in different situations

As example:

* Change the population size
* Change the number of days to simulate
* Change the first infected person

#include <stdio.h>

int day = 1, length;

char people[20];

void updateRecovered() {

// TODO Almost infected people (A) becomes recovered (R)

for(int i = 0; i < length; i++){

if (people[i] == 'A'){

people[i] = 'R';

}

}

}

void updateAlmostRecovered() {

// TODO Infected people (I) becomes almost infected (A)

for(int i = 0; i < length ; i++){

if (people[i] == 'I'){

people[i] = 'A';

}

}

}

void spreadVirus() {

// TODO Spread the virus, following the rules

for (int i = 0; i < length; i++){

if (people[i] == 'A'){

if(people[i + 1] == 'H'){

people[i+1] = 'I';

}

if(people[i-1] == 'H'){

people[i-1] = 'I';

}

}

}

}

void display() {

// TODO Display the population and the current day.

printf("\nThis is the data of the population day %d : ", day);

for(int i = 0; i < length; i++){

printf("%c ", people[i]);

}

}

int main() {

printf("Enter the data of people : ");

scanf("%20s", people);

for(length = 0; people[length] != '\0'; length++ );

display();

for (day = 2; day <= 20; day++) {

updateRecovered();

updateAlmostRecovered();

spreadVirus();

display();

}

printf("\nSimulation over!\n");

return 0;

}