1.1

Too large or too small values of integers may fall outside the range of the data type, leading to undefined behavior that may reduce the robustness of the code as well as give rise to security vulnerabilities.

And here we have x and y of type size_t which is not defined and this could lead to go above the upper limit which can cause underflow/overflow.

1.2

That is how we fix it:

```
#include <limits.h>
int main() {
        char *matriz;
        printf("%d\n", INT_MAX);
        vulneravel(matriz, 2147483647 +2 ,2147483647 ,1);
}
```

1.3

An error will occur which is a segmentation error because we try to allocate a size of memory that is less than what is expected which leads to have memory regions that cannot be allocated.

2.1

The problem is in the argument passed although in the if statement it is guaranteed that the range of the argument passed is between 0 and MAX_SEIZE which is defined as 2048 which makes us control how much

of memory allocation we get but it tamanho_real from the same type as tamanho and tamanho_real comes from tamanho -1 so if tamanho is equal to 0 then tamano-real will become -1 there will be and underflow because variables from type size_t does not have representation of negative numbers.

2.2

Demensostrating the problem requires passing the size argument with value 0.

```
Int main(){
        char* words = “random words\n”;
        vulneravel(words,0);
}
```

2.3

Segmentation error will occur because the desired memory allocation does not exist.

2.4

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


const int MAX_SIZE = 2048;
```

```
const int MIN_SIZE = 0;


void vulneravel (char *origem, size_t tamanho) {
size_t tamanho_real;
char *destino;

if (tamanho < MAX_SIZE && tamanho > MIN_SIZE) {
tamanho_real = tamanho - 1; // Não copiar \0 de origem para destino
destino = (char *) malloc(tamanho_real);
memcpy(destino, origem, tamanho_real);
}


}


int main() {
vulneravel("random words", 0);
}
```

We need to validate that there will be no negative values and not go above the maximum value so we do that by changing the if statement from

If (tamanho < MAX_SIZE)

to

If (tamanho < MAX_SIZE && tamanho > MIN_SIZE)

Where we defined MIN_SIZE as the value 0.