Parte Teórica

1) Corregir los errores de sintaxis en el código siguiente:

```
#include <stdio.h>
int main(void){
	int i;		//defino variable tipo int
	int &pi;	//defino puntero a int
	pi = &i;	//inicializo puntero
	*pi = 1;	//inicializo variable

	incrementar(*i); //incremento variable
	incrementar(&pi);	//incremento variable usando el puntero
}
void incrementar(int * nro){
	&nro = &nro + 1;
}
```

2) El prototipo de la función memset es el siguiente:

```
void *memset(void *s, int c, size_t n);
```

La función memset() rellena los primeros n bytes del área de memoria apuntada por s con el byte constante c.
Cuál es la forma correcta de invocar a esta función para que inicialice todas sus posiciones con un valor constante?

```
#include <stdio.h>
#define SIZE 10
#define INIT_VALUE 0
int main(void){
	int array[SIZE];
	memset(&array,INIT_VALUE,sizeof(int) * SIZE);		//a
	memset((void *)array,INIT_VALUE,sizeof(int) * SIZE); //b
	memset(*array,INIT_VALUE,sizeof(int) * SIZE);		//c
	memset(array,INIT_VALUE,sizeof(int) * SIZE);		//d
	memset(int array,INIT_VALUE,sizeof(int) * SIZE);	//e
	memset(array[],INIT_VALUE,sizeof(int) * SIZE);		//f
	memset(array[10],INIT_VALUE,sizeof(int) * SIZE);	//g
}
```

3) Explicar por qué cuando se compila código que invoca la función gets se obtiene un warning.

char *gets(char *s)

gets() lee una línea de stdin y la guarda en el búfer al que apunta s hasta que se encuentre bien un carácter terminador nueva-línea, bien EOF, al cual reemplaza con '\0'.

Parte Práctica

Nota: En cada ejercicio realizar, adicionalmente a la función que se requiere, un main que la invoque correctamente.

1) Realizar una función que reemplace todas las ocurrencias de un caracter dado dentro de un string por otro caracter.

int replace_char(char *str, char buscado, char reemplazo);
La función devuelve la cantidad de caracteres reemplazados.

2) Realizar una función que elimine de un string las ocurrencias de un caracter dado.

int delete_char(char *str, char buscado);

La función devuelve la cantidad de caracteres eliminados.