

1. Punteros

Ejercicio 1.1. El siguiente programa compila y corre sin errores. Indicar qué es lo que escribe y justifique su respuesta.

```
1  #include <stdio.h>
2
3  void acumular(int * x, int * y);
4
5  void main(void)
6  {
7      int x = 20;
8      int y = 10;
9
10     acumular(&x, &y);
11     printf("\n x: %d, y: %d\n", x, y);
12     acumular(&y, &x);
13     printf(" x: %d, y: %d\n\n", x, y);
14 }
15
16 void acumular(int * x, int * y)
17 {
18     *x = *y + *x;
19 }
```

Ejercicio 1.2. Indique lo que se mostrará en la pantalla al ejecutar cada programa y explique por qué.

1. `#include<stdio.h>`

```
int main()
{
    int i=3, *j, k;
    j = &i;
    printf(" %d\n", i**j*i**j);
    return 0;
}
```

2. `#include<stdio.h>`

```
int main()
{
```

```

    int x = 30, *y, *z;
    y = &x;
    z = y;
    *y++ = *z++;
    x++;
    printf(" x = %d, y = %d, z = %d\n", x, y, z);
    return 0;
}

```

3. `#include<stdio.h>`

```

int main()
{
    int ***r, **q, *p, i = 8;
    p = &i;
    q = &p;
    r = &q;
    printf(" %d, %d, %d\n", *p, **q, ***r);
    return 0;
}

```

Ejercicio 1.3. ¿Qué se imprime y por qué?. Corregir si es necesario.

`#include <stdio.h>`

```

int main()
{
    char *pc = NULL;
    int *pi = NULL;
    double *pd =NULL;
    printf("\n%d %d %d\n%d %d %d\n\n", (int)(pc +1), (int)(pi +1), (int)(pd +1), (int)
}

```

Ejercicio 1.4. La criba de Eratstenes es un eficiente algoritmo para calcular los nmeros primos. El primer paso del algoritmo es anotar todos los nmeros desde el 2 hasta una lmite dado. En el resto del algoritmo, se recorre la lista y se borra los nmeros que no son primos. Aqu los pasos restantes. Empezando en el inicio de la lista, encontramos el primer nmero que no ha sido tachado, el cul debe ser el nmero 2. Tachamos cada segundo nmero de la lista despues del 2, ya que ellos son mltiplos de 2. Entonces repetimos esos pasos desde el inicio de la lista. El primer nmero de la lista que no es tachado es ahora el 3, as que tachamos cada tercer nmero despues del 3. El prximo nmero en la lista es el 4, que fue ya tachado, as que lo saltamos. Cuando tu finalizas este proceso, los nmeros

que no son tachados, son primos. Escribe un programa que implemente este algoritmo, usando un arreglo, para tu lista. El valor de cada elemento de la matriz realiza un seguimiento de si el número correspondiente se ha tachado. Ajusta todos los elementos de la matriz a TRUE inicialmente, y luego ir poniendo FALSE según el algoritmo anterior vaya eliminando. Nota que ninguno de los números pares son primos excepto el 2. Puedes hacer el programa más eficiente haciendo un seguimiento solo de los números impares en el arreglo.