1)

```c
#include <stdio.h>
int main(void) {
  printf("Ingrese los valores para x, y, z, separando con espacios\n");
  int x, y, z;
  scanf("%d %d %d", &x, &y, &z);

  char expresiones[5][30] = {
    "x+y+1", "z*z+y*45-15*x", "y-2 == (x*3+1) % 5", "y/2*x", "y<x*z"
  };
  int valores[5] = {x+y+1, z*z+y*45-15*x, y-2 == (x*3+1) % 5, y/2*x, y<x*z};

  for (int i=0;i<5;i++) {
    printf("%s --> %d\n", expresiones[i], valores[i]);
  };

  return 0;
}
```

Expresion	$(x \mapsto 7, y \mapsto 3, z \mapsto 5)$	$(x \mapsto 1, y \mapsto 10, z \mapsto 8)$
x+y+1	11	12
z* z+y* 45-15*x	55	499
y- 2==(x*3+1)%5	0	0
y/2 * x	7	5
y < x*z	1	0

El tipo que tiene el resultado en la ultima expresion en C es Int.

2)

```c
#include <stdio.h>
#include <stdbool.h>
int main(void) {
  int x, y, z;
  bool b, w;
  int temp_b, temp_w;

  char expresiones[3][35] = {
    "x % 4 == 0                      ",
    "x + y == 0 && y - x == (-1) * z",
    "not b && w                      "
  };
  int valores[3] = {
    x % 4 == 0,
    x + y == 0 && y - x == (-1) * z,
    ! b && w
  };

  printf("Ingrese los valores para x, y, z, b y w separando con espacios\n");
  scanf("%d %d %d %d %d", &x, &y, &z, &temp_b, &temp_w);
  b = temp_b;
  w = temp_w;

  printf("\n(x->%d)\n", x);
  printf("%s --> %d\n", expresiones[0], valores[0]);

  printf("\n(x->%d, y->%d, z->%d)\n", x, y, z);
  printf("%s --> %d\n", expresiones[1], valores[1]);

  printf("\n(b->%d, w->%d)\n", b, w);
  printf("%s --> %d\n", expresiones[2], valores[2]);

  return 0;
}
```

Resultado: $(x \mapsto 4, y \mapsto -4, z \mapsto 8, b \mapsto 1, w \mapsto 0)$.

3) a)

```c
#include <stdio.h>
void uno_a(void) {
  int x;
  printf("\nEjercicio 1a: Ingrese el valor para x\n");
  scanf("%d", &x);
  x = 5;
  printf("(x->%d)\n", x);
}
void uno_b(void) {
  int x, y;
  printf("\nEjercicio 1b: Ingrese el valor para x e y (separandolos con espacios)\n");
  scanf("%d %d", &x, &y);
  x = x + y;
  y = y + y;
  printf("(x->%d, y->%d)\n", x, y);
}
void uno_c(void) {
  int x, y;
  printf("\nEjercicio 1c: Ingrese el valor para x e y (separandolos con espacios)\n");
  scanf("%d %d", &x, &y);
  y = y + y;
  x = x + y;
  printf("(x->%d, y->%d)\n", x, y);
}
int main(void) {
  uno_a();
  uno_b();
  uno_c();
  return 0;
}
```

Programa	Usuario ingresa un σ_0	Produce una salida σ
1.a Ejecucion 1	$(x \mapsto 1)$	$(x \mapsto 5)$
1.a Ejecucion 2	$(x \mapsto 3)$	$(x \mapsto 5)$
1.a Ejecucion 3	$(x \mapsto 5)$	$(x \mapsto 5)$
1.b Ejecucion 1	$(x \mapsto 1, y \mapsto 2)$	$(x \mapsto 3, y \mapsto 4)$
1.b Ejecucion 2	$(x \mapsto 3, y \mapsto 4)$	$(x \mapsto 7, y \mapsto 8)$
1.b Ejecucion 3	$(x \mapsto 5, y \mapsto 6)$	$(x \mapsto 11, y \mapsto 12)$
1.c Ejecucion 1	$(x \mapsto 2, y \mapsto 4)$	$(x \mapsto 10, y \mapsto 8)$
1.c Ejecucion 2	$(x \mapsto 5, y \mapsto 1)$	$(x \mapsto 7, y \mapsto 2)$
1.c Ejecucion 3	$(x \mapsto 9, y \mapsto 4)$	$(x \mapsto 17, y \mapsto 8)$

b)

```c
#include <stdio.h>
#include <assert.h>

int uno_a(void) {
  int x;
  printf("\nEjercicio 1a: Ingrese el valor para x\n");
  scanf("%d", &x);
  assert(x == 1);
  x = 5;
  printf("(x->%d)\n", x);
  return 0;
}

int uno_b(void) {
  int x, y;
  printf("\nEjercicio 1b: Ingrese el valor para x e y (separandolos con espacios)\n");
  scanf("%d %d", &x, &y);
  assert(x == 2 && y == 5);
  x = x + y;
  y = y + y;
  printf("(x->%d, y->%d)\n", x, y);
  return 0;
}

int uno_c(void) {
  int x, y;
  printf("\nEjercicio 1c: Ingrese el valor para x e y (separandolos con espacios)\n");
  scanf("%d %d", &x, &y);
  assert(x == 2 && y == 5);
  y = y + y;
  x = x + y;
  printf("(x->%d, y->%d)\n", x, y);
  return 0;
}

int main(void) {
  uno_a();
  uno_b();
  uno_c();
  return 0;
}
```

4)

a)

```c
#include <stdio.h>

int main(void){
  int x, y;
  printf("Ingrese los valores para x e y (separandolos con espacios)\n");
  scanf("%d %d", &x, &y);
  if (x >=  y) {
    x = 0;
  }
  else if (x <= y) {
    x = 2;
  }
  printf("(x->%d, y->%d)", x, y);
  return 0;
}
```

b)

```c
#include <stdio.h>

int main(void){
  int x, y, z, m;
  printf("Ingrese los valores para x,y,z y m (separandolos con espacios)\n");
  scanf("%d %d %d %d", &x, &y, &z, &m);
  if (x < y) {
    m = x;
  } else if (x >= y) {
    m = y;
  }
  printf("(x->%d, y->%d, z->%d, m->%d)\n", x, y, z, m);
  if (m >= z) {
    m = z;
  }

  printf("(x->%d, y->%d, z->%d, m->%d)", x, y, z, m);
  return 0;
}
```

$$\sigma_0 : (x \mapsto 5, y \mapsto 4, z \mapsto 8, m \mapsto 0)$$
$$\sigma_1 : (x \mapsto 5, y \mapsto 4, z \mapsto 8, m \mapsto 4)$$
$$\sigma_2 : (x \mapsto 5, y \mapsto 4, z \mapsto 8, m \mapsto 4)$$

Lo que hace el programa es determinar cual es el menor valor entre 3 enteros. El valor final de la variable m es el mismo valor que la variable con el valor más chico. En el caso del ejemplo, dicho valor es 4.

5)

b)

1)

```c
#include <stdio.h>

int main(void){
  int x, y, i;
  printf("Ingrese el valor para x, y e i (Separandolos utilizando espacios)\n");
  scanf("%d %d %d", &x, &y, &i);
  i = 0;
  while (x >= y) {
    x = x-y;
    i = i + 1;
    printf("(x->%d, y->%d, i->%d)\n", x, y, i);
  }
  return 0;
}
```

$$\sigma_0 : (x \mapsto 13, y \mapsto 3, i \mapsto 0)$$

$$\sigma_1^0 : (x \mapsto 10, y \mapsto 3, i \mapsto 1)$$
$$\sigma_1^1 : (x \mapsto 7, y \mapsto 3, i \mapsto 2)$$
$$\sigma_1^2 : (x \mapsto 4, y \mapsto 3, i \mapsto 3)$$
$$\sigma_1^3 : (x \mapsto 1, y \mapsto 3, i \mapsto 4)$$

2)

```c
#include <stdio.h>
#include <stdbool.h>

int main(void){
  int x, i, temp_res;
  bool res;
  printf("Ingrese el valor para x, i, res (Separandolos utilizando espacios)\n");
  scanf("%d %d %d", &x, &i, &temp_res);
  res = temp_res;
  i = 2;
  res = true;
  while (i < x && res) {
    res = res && (x % i) != 0;
    i = i + 1;
    printf("(x->%d, i->%d, res->%d)\n", x, i, res);
  }
  return 0;
}
```

$$\sigma_0 : (x \mapsto 5, i \mapsto 0, res \mapsto False)$$

$$\sigma_1^0 : (x \mapsto 5, i \mapsto 3, res \mapsto 1)$$
$$\sigma_1^0 : (x \mapsto 5, i \mapsto 4, res \mapsto 1)$$
$$\sigma_1^0 : (x \mapsto 5, i \mapsto 5, res \mapsto 1)$$

3) Ejercicio 5b1:

$$\sigma_0 : (x \mapsto 10, y \mapsto 5, i \mapsto 0)$$

$$\sigma_1^0 : (x \mapsto 5, y \mapsto 5, i \mapsto 1)$$
$$\sigma_1^1 : (x \mapsto 0, y \mapsto 5, i \mapsto 2)$$

Lo que hace el programa es realizar la division entera de **x** por **y** y guarda el resultado en **i**.

Ejercicio 5b2:

$$\sigma_0 : (x \mapsto 21, i \mapsto 0, res \mapsto False)$$

$$\sigma_1^0 : (x \mapsto 21, i \mapsto 3, res \mapsto True)$$
$$\sigma_1^0 : (x \mapsto 21, i \mapsto 4, res \mapsto False)$$

Otro estado inicial:

$$\sigma_0 : (x \mapsto 7, i \mapsto 0, res \mapsto False)$$

$$\sigma_1^0 : (x \mapsto 7, i \mapsto 3, res \mapsto True)$$
$$\sigma_1^1 : (x \mapsto 7, i \mapsto 4, res \mapsto True)$$
$$\sigma_1^2 : (x \mapsto 7, i \mapsto 5, res \mapsto True)$$
$$\sigma_1^3 : (x \mapsto 7, i \mapsto 6, res \mapsto True)$$
$$\sigma_1^4 : (x \mapsto 7, i \mapsto 7, res \mapsto True)$$

El programa determina si **x** es un número primo o no y guarda ese resultado en **res**. Para determinar dicho resultado verifica si existe algún número mayor a 2 y menor a **x** que divida a **x**.

6)a)

```c
#include <stdio.h>
int pedirEntero(void) {
  int x;
  printf("Ingrese un numero entero\n");
  scanf("%d", &x);
  return x;
}

void imprimeEntero(int x) {
  printf("El entero es: %d", x);
}

int main(void) {
  int entero = pedirEntero();
  imprimeEntero(entero);
  return 0;
}
```

6)b)

```c
#include <stdio.h>

int pedirEntero(char *nombre){
  int x;
  printf("Ingrese un valor\n%s = ", nombre);
  scanf("%d", &x);
  return x;
}

void imprimeEntero(int x, char *nombre) {
  printf("\n%s --> %d", nombre, x);
}

int devolver_mayor(int x, int y, int z, int m){
  if (x < y) {
    m = x;
  } else if (x >= y) {
    m = y;
  } else if (m >= z) {
    m = z;
  }
  return m;
}

int main(void){
  int x, y, z, m;
  x = pedirEntero("x");
  y = pedirEntero("y");
  z = pedirEntero("z");
  m = pedirEntero("m");

  m = devolver_mayor(x, y, z, m);

  imprimeEntero(x, "x");
  imprimeEntero(y, "y");
  imprimeEntero(z, "z");
  imprimeEntero(m, "m");
  return 0;
}
```

Las ventajas que se encuentran en esta nueva version del programa son que es necesario repetir menor cantidad de codigo, además de que facilita la lectura del programa y simplifica el proceso de hacer cambios en caso de ser necesario, ya que con cambiar la definicion de la funcion es suficiente para cambiar su comportamiento en todo el programa.

Algunos ejercicios redefinidos usando funciones:

Ej1:

```c
#include <stdio.h>
int pedirEntero(char *nombre){
  int x;
  printf("Ingrese un valor\n%s = ", nombre);
  scanf("%d", &x);
  return x;
}

void imprimeEntero(int x, char *nombre) {
  printf("\n%s --> %d", nombre, x);
}

int main(void) {
  printf("Ingrese los valores para x, y, z, separando con espacios\n");
  int x, y, z;
  x = pedirEntero("x");
  y = pedirEntero("y");
  z = pedirEntero("z");

  char expresiones[5][30] = {
    "x+y+1", "z*z+y*45-15*x", "y-2 == (x*3+1) % 5", "y/2*x", "y<x*z"
  };
  int valores[5] = {x+y+1, z*z+y*45-15*x, y-2 == (x*3+1) % 5, y/2*x, y<x*z};

  for (int i=0;i<5;i++) {
    imprimeEntero(valores[i], expresiones[i]);
  };

  return 0;
 }
```

Ej2:

```c
#include <stdio.h>
#include <stdbool.h>

int pedirEntero(char *nombre){
  int x;
  printf("Ingrese un valor\n%s = ", nombre);
  scanf("%d", &x);
  return x;
}

void imprimeEntero(int x, char *nombre) {
  printf("\n%s --> %d", nombre, x);
}

int main(void) {
  int x, y, z;
  bool b, w;
  int temp_b, temp_w;

  x = pedirEntero("x");
  y = pedirEntero("y");
  z = pedirEntero("z");
  temp_b = pedirEntero("b");
  temp_w = pedirEntero("w");

  b = temp_b;
  w = temp_w;

  char expresiones[3][35] = {
    "x % 4 == 0                      ",
    "x + y == 0 && y - x == (-1) * z",
    "not b && w                      "
  };
  int valores[3] = {
    x % 4 == 0,
    x + y == 0 && y - x == (-1) * z,
    ! b && w
  };

  imprimeEntero(x, "x");
  imprimeEntero(valores[0], expresiones[0]);

  imprimeEntero(x, "x");
  imprimeEntero(y, "y");
```

```
 imprimeEntero(z, "z");
 imprimeEntero(valores[1], expresiones[1]);

 imprimeEntero(b, "b");
 imprimeEntero(w, "w");
 imprimeEntero(valores[2], expresiones[2]);

 return 0;
}
```

Ej3b:

```c
#include <stdio.h>
#include <assert.h>

int pedirEntero(char *nombre){
  int x;
  printf("Ingrese un valor\n%s = ", nombre);
  scanf("%d", &x);
  return x;
}

void imprimeEntero(int x, char *nombre) {
  printf("\n%s --> %d\n", nombre, x);
}

int uno_a(void) {
  int x;
  x = pedirEntero("x");
  assert(x == 1);
  x = 5;
  imprimeEntero(x, "x");
  return 0;
}

int uno_b(void) {
  int x, y;
  x = pedirEntero("x");
  y = pedirEntero("y");
  assert(x == 2 && y == 5);
  x = x + y;
  y = y + y;
  imprimeEntero(x, "x");
  imprimeEntero(y, "y");
  return 0;
}

int uno_c(void) {
  int x, y;
  x = pedirEntero("x");
  y = pedirEntero("y");
  assert(x == 2 && y == 5);
  y = y + y;
  x = x + y;
  imprimeEntero(x, "x");
  imprimeEntero(y, "y");
```

```
  return 0;
}

int main(void) {
  uno_a();
  uno_b();
  uno_c();
  return 0;
}
```

6)c)

```
#include <stdio.h>

void imprimeHola(void){
  printf("Hola\n");
}

void imprimeChau(void){
  printf("Chau\n");
}

int main(void) {
  imprimeHola();
  imprimeHola();
  imprimeChau();
  imprimeChau();
  return 0;
}
```