

**Ejercicio 1:**

```
int *punt;  
int x = 7;  
int y = 5;  
punt = &x;  
*punt = 4;  
printf("%d, %d", x, y);
```

Resultado: 4, 5

**Ejercicio 2:**

```
int *punt;  
int x = 7;  
int y = 5;  
punt = &x;  
x = 4;  
printf("%d, %d", *punt, y);
```

Resultado: 4, 5

**Ejercicio 3:**

```
int *punt;  
int x = 7;  
int y = 5;  
punt = &x;  
x = 4;  
punt = &y;  
printf("%d, %d", *punt, x);
```

Resultado: 5, 4

**Ejercicio 4:**

```
int *punt;  
int x = 7;  
int y = 5;  
punt = &x;  
*punt = 3;  
punt = &y;  
*punt = x;  
x = 9;  
printf("%d, %d", *punt, x);
```

Resultado: 7, 9

**Ejercicio 5:**

```
int *punta, *puntb;  
int x = 7;  
int y = 5;  
punta = &x;  
*punta = 3;  
puntb = &y;  
*puntb = x;  
x = 9;  
printf("%d, %d", *puntb, x);
```

Resultado: 9, 9

**Ejercicio 6:**

```
int *punta, *puntb;  
int x = 7;  
int y = 5;  
punta = &x;  
*punta = 3;  
puntb = &y;
```

```
*puntb = x;  
x = 9;  
printf("%d, %d", *puntb, *punta);
```

Resultado: 9, 9

#### Ejercicio 7:

```
int *punta, *puntb;  
int x = 7;  
int y = 5;  
punta = &x;  
*punta = 3;  
puntb = &y;  
*puntb = x;  
x = 9;  
puntb = punta;  
printf("%d, %d", *puntb, y);
```

Resultado: 3, 5

#### Ejercicio 8:

```
int *punt, i;  
int x[5] = {1, 2, 3, 4, 5};  
punt = x;  
*punt = 9;  
for (i = 0; i < 5; i++)  
    printf("%d,", x[i]);
```

Resultado: 9,9,3,4,5,

#### Ejercicio 9:

```
int *punt, i;  
int x[5] = {1, 2, 3, 4, 5};  
punt = &x[0];  
*punt = 9;  
punt[3] = 7;  
for (i = 0; i < 5; i++)  
    printf("%d,", x[i]);
```

Resultado: 9,2,3,7,5,

#### Ejercicio 10:

```
int *punt, i;  
int x[5] = {1, 2, 3, 4, 5};  
punt = x;  
*x = 11;  
*(punt + 3) = 9;  
for (i = 0; i < 5; i++)  
    printf("%d,", x[i]);
```

Resultado: 11,2,3,9,5,

#### Ejercicio 11:

```
int *punt, i;  
int x[5] = {1, 2, 3, 4, 5};  
punt = x;  
*(punt + 2) = 9;  
*(x + 3) = 7;  
punt[1] = 11;  
for (i = 0; i < 5; i++)  
    printf("%d,", *(punt + i));
```

Resultado: 1,11,9,7,5,

#### Ejercicio 12:

```

int *punt, i;
int x[5] = {1, 2, 3, 4, 5};
punt = x + 4;
*(punt - 2) = 9;
punt--;
*(punt) = 7;
punt[1] = 11;
for (i = 0; i < 5; i++)
    printf("%d,", *(x + i));

```

Resultado: 1,9,7,11,5,

### Ejercicio 13:

```

int *punt, i;
int x[5] = {1, 2, 3, 4, 5};
punt = &x[0] + 3;
*(punt - 2) = 9;
punt--;
*(punt) = 7;
punt[1] = 11;
punt = x;
for (i = 0; i < 5; i++)
    printf("%d,", punt[i]);

```

Resultado: 9,7,11,4,5,

### Ejercicio 14:

```

void suma_dos(int *x, int *y, int *z)
{
    *x = *x + 2;
    *y = *y + 2;
    *z = *z + 2;
}
void main(void)
{
    int x, y, z;
    x = 3;
    y = 10;
    z = 15;
    suma_dos(&x, &y, &z);
    printf("%d %d %d", x, y, z);
}

```

Resultado: 5 12 17

### Ejercicio 15:

```

void datos(int *x, float *y, char *c)
{
    *x = 8;
    *y = 4.2;
    *c = 'g';
}
void main(void)
{
    int x = 9;
    float y = 44.6;
    char c = 'a';
    datos(&x, &y, &c);
    printf("%d %f %c", x, y, c);
}

```

Resultado: 8 4.200000 g

**Ejercicio 16:**

```
void datos(int *x, float *y, char *c)
{
    printf("%d %f %c", *x, *y, *c);
    *x = 8;
    *y = 4.2;
    *c = 'g';
}
void main(void)
{
    int x = 9;
    float y = 44.6;
    char c = 'a';
    datos(&x, &y, &c);
    printf("%d %d %f %c", x, y, c);
}
```

Resultado: 9 44.599998 a

**Ejercicio 17:**

```
void datos(int x, float y, char c)
{
    printf("%d %f %c", x, y, c);
    x = 8;
    y = 4.2;
    c = 'g';
}
void main(void)
{
    int x = 9;
    float y = 44.6;
    char c = 'a';
    datos(x, y, c);
    printf("%d %d %f %c", x, y, c);
}
```

Resultado: 9 44.599998 a

**Ejercicio 18:**

```
int datos(int x, float y, char c)
{
    printf("%d %f %c", x, y, c);
    x = 8;
    y = 4.2;
    c = 'g';
    return x;
}
void main(void)
{
    int x = 9;
    float y = 44.6;
    char c = 'a';
    x = datos(x, y, c);
    printf("%d %d %f %c", x, y, c);
}
```

Resultado: 9 44.599998 a

**Ejercicio 19:**

```
char datos(int *x, float *y, char *c)
{
    printf("%d %f %c", *x, *y, *c);
    *x = 8;
    *y = 4.2;
    *c = 'g';
    return 'h';
}
```

```
}  
void main(void)  
{  
    int x = 9;  
    float y = 44.6;  
    char c = 'a';  
    c = datos(&x, &y, &c);  
    printf("%d %f %c", x, y, c);  
}
```

Resultado: 8 4.200000 g