1) CALCULAR EL ÁREA DE UN TRIÁNGULO =

```
INT A;
INT B;
INT RESULTADO;
// INGRESAR, GUARDAR Y MOSTRAR LA BASE Y LA ALTURA
PRINTF ("VALOR DE ALTURA\n");
SCANF ("%d", &A);
PRINTF ("VALOR DE BASE\n");
SCANF ("%d", &B);
// CALCULAR EL ÁREA
RESULTADO = A x B;
PRINTF ("EL ÁREA DEL TRIÁNGULO ES: %d\n", RESULTADO);
RETURN 0;
```

P.C:

→ VALOR DE ALTURA
→ 5
→ VALOR DE BASE
→ 6
(5×6 =30)
→ EL ÁREA DEL TRIÁNGULO ES: 30

2) INGRESAR UN NÚMERO FIJ INDICAR SI ES PAR O IMPAR =

P.C:

```
INT A;
INT RESULTADO;
// INGRESAR UN NÚMERO PARA SABER SI ES PAR O IMPAR
PRINTF ("VALOR DE LA VARIABLE\n");
SCANF ("%d", &A);
RESULTADO = A/2;
// PAR
IF (RESULTADO == 0) {
    PRINTF ("LA VARIABLE ES PAR");
}
ELSE {
    PRINTF ("LA VARIABLE ES IMPAR");
}
RETURN 0;
```

→ VALOR DE VARIABLE
→ 4
(4/2 =2 ES PAR)
→ LA VARIABLE ES PAR

4) PEDIR UN NÚMERO DEL 1 AL 7 Y Q' DIGA EL DÍA DE LA SEMANA CORRESPONDIENTE =

P.C:

```
INT I;
PRINTF ("INGRESE VALOR DEL 1 AL 5\n");
SCANF ("%d", &I);

SWITCH (I) {
    CASE 1:
        PRINTF ("LUNES\n");
        BREAK;
    CASE 2: PRINTF ("MARTES\n");
        BREAK;
    CASE 3: PRINTF ("MIÉRCOLES\n");
        BREAK;
    CASE 4: PRINTF ("JUEVES\n");
        BREAK;
    CASE 5: PRINTF ("VIERNES\n");
        BREAK;
    DEFAULT: PRINTF ("OPCIÓN NO VÁLIDA\n");
        BREAK;
```

→ INGRESA VALOR DEL 1 AL 5
→ 4
→ JUEVES

5) Realizar un contador de 0 a 5 =

```
int I=0;
printf("contador\n");
for(I=0; I<=5; I++){
    printf("%d", I);
}
return 0;
```
→ contador
→ 012345   Pc:

6) Ingresar un valor inicial y final y q' me cuente los números dentro de sus valores límites.

```
int I,F,k;
printf("Ingrese un valor inicial\n");
scanf("%d",&I);
printf("Ingrese un valor final\n");
scanf("%d",&F);
for(k=inicial; k<=final; k++){
    printf("%d",&k);
}
return 0;
```
→ Ingrese un valor inicial
→ 1
→ Ingrese un valor final
→ 10
→ 1 2 3 4 5 6 7 8 9 10

7) Contador de 0 a 5 =

```
int I=0;
printf("contador\n");
while(I<=5){
    I++;
    printf("%d", I);
}
return 0;
```
Pc:
0
→ contador
0+1...
→ 012345

8) Sumar los primeros 20 números =

```
int n=0;
int S=0;

while(n<=20){
    S=S+n;
    n++;
}
printf("%d\n", S);
return 0;
```
Pc:
0
0
→ 0+1
→ 0+1
→ 210

RETURN 0;

10) ARMAR UNA FIGURA CUADRADA =

INT I, J;

```
FOR (I=1 ; I <= 3 ; I++) {
    FOR (J=1 ; J <= 5 ; J++) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}

RETURN 0;
```

P C :

```
      5
    # # # # #
    # # # # #   3
    # # # # #
```

11) ARMAR UNA FIGURA TRIANGULAR CON 1 VARIABLE =

INT I, J, K, TAMAÑO;

```
PRINTF ("INGRESE TAMAÑO : ");
SCANF ("%d", &TAMAÑO);

FOR (I=1 ; I <= TAMAÑO ; I++) {
    FOR (J=1 ; J < TAMAÑO ; J++) {
        PRINTF (" ")
    }
    FOR (K=1 ; K <= I ; K++) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}
RETURN 0;
```

INGRESE TAMAÑO : 10

```
            #          ESPACIOS = 3
           # #
          # # #
         # # # #
        # # # # #        10 (TAMAÑO)
       # # # # # #
      # # # # # # #
     # # # # # # # #
    # # # # # # # # #
   # # # # # # # # # #
```

I

12) MISMO q' EL ANTERIOR PERO AL REVES =

```
INT I, J, K, Tamaño;

PRINTF ("INGRESE TAMAÑO: ");
SCANF ("%d", & Tamaño);

FOR (I = Tamaño; I >= 1; I--) {
    FOR (J = 1; J <= Tamaño; J++) {
        PRINTF (" ");
    }
    FOR (K = I; K >= 1; K--) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}
```

PC:

INGRESE TAMAÑO: 5

```
I   # # # # # _
    # # # # _ _
    # # # _ _ _      Tamaño ES 5
    # # _ _ _ _
K   # _ _ _ _ _
```

13) ARMAR 1 FIGURA PIRAMIDAL CON 1 VARIABLE =

```
INT I, J, K, Tamaño;

PRINTF ("INGRESE TAMAÑO: ");
SCANF ("%d", & Tamaño);

FOR (I = 1; I <= Tamaño; I++) {
    FOR (J = 1; J <= Tamaño - I; J++) {
        PRINTF (" ");
    }
    FOR (K = 1; K <= (2*I) - 1; K++) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}
RETURN 0;
```

PC:

INGRESE TAMAÑO: 5

```
          I
          #
        # # #        x2 ES K
      # # # # #
    # # # # # # #          Tamaño: 5
  # # # # # # # # #
```

14) MISMO q' EL ANTERIOR PERO AL REVES =

```
INT I, J, K, Tamaño;

PRINTF ("INGRESE TAMAÑO: ");
SCANF ("%d", & Tamaño);

FOR (I = Tamaño; I >= 1; I--) {
    FOR (J = 1; J <= Tamaño; J++) {
        PRINTF (" ");
    }
    FOR (K = (2*I) - 1; K >= 1; K--) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}
RETURN 0;
```

PC:

INGRESE TAMAÑO: 5

```
# # # # # # # # #
  # # # # # # #
    # # # # #            Tamaño ES 5
      # # #        2x ES K
        #
```

15) Realizar la Figura RECTANGULO HUECO con VARIABLE =

```
INT I, J, k, Tamaño;

PRINTF ("Ingresar Tamaño: ");
scanf ("%d", &Tamaño);

FR (I=1; I <= Tamaño; I++) {
    IF (I == 1 || I == Tamaño) {
        FR (k=1; k <= 10; k++) {
            PRINTF ("#");
        }
    }
    IF (I > 1 && I < Tamaño) {
        PRINTF ("#");
        FR (J=1; J <= 10-2; J++) {
            PRINTF (" ");
        }
        PRINTF ("#");
    }
    PRINTF ("\n");
}
RETURN 0;
```

PC:

```
Ingresar Tamaño: 4
            10 = k
  I  #HHHHHHHHHHHHHHHHH
Tamaño  #  J = 10-2 (IF)
  4  #
     HHHHHHHHHHHHHHHHHH
```

16) TRIANGULO 1 SOLO ASTERISCO =

```
INT I, J;

FR (I=1; I <= 5; I++) {
    FR (J=1; J <= I; J++) {
        PRINTF (" ");
    }
    PRINTF ("+\n");
}
RETURN 0;
```

(+\n) PC:

```
      *
    - +      I
    - - +
    - - - +
    - - - - +
```

17) LETRA C =

```
INT I, J;

FR (I=1; I <= 2; I++) {
    FR (J=1; J <= 6; J++) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}

FR (I=1; I <= 2; I++) {
    FR (J=1; J <= 2; J++) {
        PRINTF ("#");
    }
    PRINTF ("\n")
}
```

```
FR (I=1; I <= 2; I++) {
    FR (J=1; J <= 6; J++) {
        PRINTF ("#");
    }
    PRINTF ("\n");
}
RETURN 0;
```

PC:

```
  I  [ # # # # # #  ] J 1-6
     [ # # # # # #  ]
     # #  ] J 1-2
     # #
     # # # # # #  J 1-6
     # # # # # #
```

18) REALIZAR UN ALGORITMO q' PIDA EL INGRESO DE 3 NÚMEROS ENTEROS, MOSTRAR POR PANTALLA EL MÁS CHICO Y EL MÁS GRANDE. MOSTRAR POR PANTALLA LOS NÚMEROS ENTRE EL MÁS CHICO Y EL MÁS GRANDE. Y REALIZAR PRUEBA DE ESCRITORIO.

```c
INT N1, N2, N3, MAX = -9999, MIN = 9999, I = 0;
PRINTF ("Ingrese N1\n");
SCANF ("%d", &N1);
PRINTF ("Ingrese N2\n");
SCANF ("%d", &N2);
PRINTF ("Ingrese N2\n");
SCANF ("%d", &N3);

IF (N1 <= MIN) {
      MIN = N1;
}
IF (N2 <= MIN) {
      MIN = N2;
}
IF (N3 <= MIN) {
      MIN = N3;
}
IF (N1 >= MAX) {
      MAX = N1
}
IF (N2 >= MAX) {
      MAX = N2
}
IF (N3 >= MAX) {
      MAX = N3
}
FOR (I = MIN; I <= MAX; I++) {
      PRINTF ("%d", &I);
}

RETURN 0
```

PC:

INGRESE N1
8
INGRESE N2
-4
INGRESE N3
2

$8 \leq$ MIN (9999)
MIN = 8

$-4 \leq 8$
MIN = -4

$2 \leq -4$
-4 (MIN)

$8 \geq -9999$
MAX = 8

$-4 \geq 8$
MAX = 8

$2 \geq 8$
8 (MÁX)

-4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8.

P. RECALEROL

INSTRUCCIONES	I	N1	N2	N3	MIN	MAX	CONSOLA:
		0	0	0	9999	-9999	—
Inicio de variables —		0	0	0	9999	-9999	INGRESE N1°
PRINT y SCAN N1 —		8	0	0	9999	-9999	INGRESE N2
PRINT y SCAN N2 —		0	-4	0	9999	-9999	INGRESE N3
PRINT y SCAN N3 —		0	0	2	9999	-9999	
IF N1 <= MIN —		8	-4	2	-4	-9999	—
IF N2 <= MIN —		8	-4	2	-4	-9999	—
IF N3 <= MIN —		8	-4	2	-4	-9999	"MIN" 4
IF N1 >= MAX —		8	-4	2	-4	8	—
IF N2 >= MAX —		8	-4	2	-4	8	—
IF N3 >= MAX —		8	-4	2	-4	8	"8" MAX
INICIO DE I	0	8	-4	2	-4	8	—
FOR I=MIN ; I<=MAX	-4	8	-4	2	-4	8	"-4"
//	-3	8	-4	2	-4	8	"-4,-3"
//	-2	8	-4	2	-4	8	"-4,-3,-2"
//	-1	8	-4	2	-4	8	"-4,-3,-2,-1"
//	0	8	-4	2	-4	8	"-4,-3,-2,-1, 0"
//	1	8	-4	2	-4	8	"-4,-3,-2,-1,0 1"
//	2	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2"
//	3	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2,3"
//	4	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2,3,4"
//	5	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2,3,4,5"
//	6	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2,3,4,5,6"
//	7	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2,3,4,5,7"
FIN FOR	8	8	-4	2	-4	8	"-4,-3,-2,-1,0 1,2,3,4,5,7,8"

19) MENU DE OPCIONES; INGRESAR 4 OPCIONES : DONDE 1 SUMA, 2 PRODUCTO, 3 EL VALOR MIN

Y 4 COMPRENDIDOS ENTRE MIN Y MAX, 0 PARA CERRAR PROGRAMA. PIDA 1ro 2 NÚMEROS ENTEROS.

(MIENTRAS NO SEA 0 ANDARÁ EL PROGRAMA)

```
INT N1, N2, I, SUMA, PRODUCTO, MIN = 9999, MAX = -9999;

PRINTF(" INGRESE PRIMER NÚMERO\n");
SCANF("%d", &N1);
PRINTF(" INGRESE SEGUNDO NÚMERO\n");
SCANF("%d", &N2);
WHILE (I!=0){
    PRINTF("1. SUMA DE LOS NÚMEROS");
    PRINTF("2. PRODUCTO " " " " \n");
    PRINTF("3 . VALOR MINIMO\n");
    PRINTF("4 . COMPRENDIDO ENTRE LOS NÚMEROS\n");
    PRINTF(" 0. CIERRE DEL PROGRAMA\n");
    SCANF("%d", &I);
IF (N1 <= MIN){
        MIN = N1;
    }
IF (N2 <= MIN){
    MIN = N2;
    }
IF (N1 >= MAX){
    MAX = N1;
    }
IF (N2 >= MAX){
    MAX = N2;
    }
SWITCH (I){
    CASE 1: SUMA = N1 + N2;
            PRINTF(" LA SUMA ES: %d", SUMA);
            BREAK;
    CASE 2: PRODUCTO = N1 * N2;
            PRINTF(" LA MULTIPLICACIÓN ES: %d", PRODUCTO);
            BREAK;
    CASE 3: PRINTF(" EL MÍNIMO ES: %d", MIN);
            BREAK;
    CASE 4: PRINTF(" LOS COMPRENDIDOS : ");
            INT F = 0;
            FOR (F = MIN; F <= MAX; F++){
                PRINTF("%d", &F);
                }
            BREAK;
    CASE 0: RETURN 0;
            BREAK;
    DEFAULT: PRINTF(" OPCIÓN NO VÁLIDA");
            BREAK;
    }
}
```

2c) Mismo q' el anterior solo q' se pedira las 4 figuras (cuadrado, triangulo, invertido, piramide)
invertida) (fuerte caso: si ingreso un numero negativo ¡no podra mostrar figura)
(y si caben con las misma la inversa su filas y columnas)

```c
int main (cuadrado, cuadradito
while (main != 0) {
        printf ("1. cuadrado");
        printf ("2. triangulo");
        printf ("3.    " invertido);
        printf ("4. piramide);
        printf (" 0. cierre del programa");
        scanf ("%d", &main);
if (main <= -1) {
        printf (" no se puede mostrar una figura");
        return 0;
    }
printf (" FILAS: ");
scanf ("%d", &cuadrado);
printf (" COLUMNAS: ");
scanf ("%d", & cuadradito);
switch (main) {
    case 1: printf (" la figura cuadrada");
        int c, cc;
        for (c = 1; c <= cuadrado; c++) {
            for (cc = 1; cc <= cuadradito; cc++) {
                printf (" #");
            }
            printf ("\n");
        }
        break;
        .
        .
        .
        .

    case 0: return 0;
        break;
    default: printf (" opción no válida");
        break;
    }
```