

Modelo 1: Solución

1. Realice la corrida en frio del siguiente programa (por lo menos 4 iteraciones) e indique su salida.

```

for (i=0;i<4;i++){
    for(j=0;j<4;j++){
        if(j<i)
            printf("* ");
        else
            if(j==i)
                printf("- ");
            else
                printf("+ ");
        printf("\n");
    }
}
    
```

i	i<4?	j	j<4?	j<i?	j==i?	Salida
0	0<4? V	0	0<4?V	0<0? F	0==0? V	-
		1	1<4? V	1<0? F	1==0?F	+
		2	2<4?V	2<0?F	2==0?F	+
		3	3<4?V	3<0?F	3==0?F	+
		4	4<4?F			\n
1	1<4?V	0	0<4?V	0<1?V		*
		1	1<4?V	1<1?F	1==1?V	-
		2	2<4?V	2<1?F	2==1?F	+

Salida general:

-+++  
 \* - +

Si seguimos haciendo la corrida en frio:

- + + +  
 \* - + +  
 \* \* - +  
 \* \* \* -

2. Escribir un programa que solicite un valor N al usuario y muestre por pantalla todos los números perfectos menores que N. Se dice que un número es perfecto cuando es igual a la suma de todos sus divisores, exceptuando el propio número. Por ejemplo: 6 es perfecto =  $1 + 2 + 3$ , 8 no es perfecto =  $1 + 2 + 4 = 7$   
28 es perfecto =  $1 + 2 + 4 + 7 + 14$ . Realizar Análisis E-P-S y Algoritmo o Codificación.

Análisis:

Entrada: Valor N que indica fin del intervalo a examinar (N) entero

Proceso:

Se dice que un número es perfecto cuando es igual a la suma de todos sus divisores excepto el mismo:

ejemplo:  $6 = 1 + 2 + 3$

Por tanto para saber si un número es perfecto, debemos calcular los divisores de dicho número y sumarlos, si la sumatoria de sus divisores es igual al número, el número será perfecto.

Cómo calcular los divisores de un número?

Un número es divisor de otro si el residuo de su división es cero.

Para encontrar todos los divisores de un número, debemos verificar para que valores se cumple que el residuo de la división sea cero. Ejemplo:

Divisores de 6

$\text{residuo}(6/1)=0$      $\text{residuo}(6/2)=0$      $\text{residuo}(6/3)=0$      $\text{residuo}(6/4)=2$   
 $\text{residuo}(6/5)=1$      $\text{residuo}(6/6)=0$

El residuo de la división es cero para 1,2,3,6

Para saber si un número es perfecto, debemos sumar los divisores excluyendo el propio número. Esto se reduce a:

ir desde  $i=1$  a  $\text{numero}-1$ , y si el  $\text{residuo}(\text{numero}/i)$  es cero, acumulamos a  $i$

Si al finalizar el acumulador de divisores es igual al numero, tenemos un número perfecto y por tanto debemos mostrarlo por pantalla.

Esto se debe repetir N veces.

Salida: Mensaje por pantalla indicando numeros perfectos (numero)

Algoritmo:

```
1 Inicio
2 Escribir("Inserte el límite: ")
3 Leer(N)
4 Para(numero=1;numero<N;numero=numero+1)
    4.1 suma_divisores=0 --> Inicializo el acumulador
    4.2 Para(i=1;i<numero;i=i+1) -->Para cada numero entre 1 y N, se deben verificar sus divisores
        4.2.1 Si (residuo(numero/i)=0) entonces
            4.2.1.1 suma_divisores=suma_divisores+i --> sumo si es divisor
            Fin_si
        Fin_RP
    4.3 Si(suma_divisores=numero) entonces --> si la sumatoria es igual al numero
        4.3.1 Escribir(numero) lo escribo por pantalla(perfecto)
        Fin_si
    Fin_RP
5. Fin
```

La estructura resaltada en amarillo, sirve para repetir la verificación de números perfectos entre 1 y N.

3. Los alumnos de cierta institución son clasificados según su índice académico (entre 0 y 1), si tienen un índice superior a 0.5 son alumnos regulares, y si el índice es superior a 0.9 son alumnos de buen rendimiento. Escribir un programa en C que lea desde teclado el número de alumnos del instituto, así como la edad e índice académico para cada alumno. Su programa debe calcular la edad media de todos los alumnos ingresados al sistema, así como la edad media de los alumnos regulares y la edad media de los alumnos de buen rendimiento. Note que los alumnos de buen rendimiento también son considerados como alumnos regulares, puesto que su índice es superior a 0.5. Realizar Análisis E-P-S y algoritmo o codificación. RECUERDE VALIDAR DATOS EN AMBOS EJERCICIOS

Análisis:

Entrada:

Número de alumnos del instituto (N) , entero

edad de alumnos (edad), entero

índice académico (índice), real

Proceso:

Para N estudiantes se debe solicitar su edad e índice.

-Para calcular la edad promedio de los alumnos ingresados, interesa llevar la sumatoria de todas las edades(edadmedia) y dividirla entre el número de estudiantes ingresados(N)

-Si el índice>0.5 --> el alumno es regular y como necesito la edad media de los alumnos regulares, debería sumar las edades (edad\_regulares)y contar el número de estudiantes regulares (num\_regulares).

-Si el índice >0.9 --> el alumno es buen rendimiento y como necesito la edad media de los alumnos de buen rendimiento, debería sumar las edades (edad\_buenr) y contar el número de estudiantes buen rendimiento (num\_buenr).

Salida:

edad media de los alumnos inscritos (edadmedia/N)  
edad media de los alumnos regulares (edad\_regulares/num\_regulares)  
edad media de los alumnos buen rendimiento (edad\_buenr/num\_buenr)

Codificación:

```
#include<stdio.h>
int main(){
    int N,edad,i,num_regulares=0,num_buenr=0;
    float indice,edadmedia=0,edad_regulares=0,edad_buenr=0;
    // el do-while hace la validación de los datos, si N<=0, se repite la lectura de
datos.
    do{
        printf("Numero de estudiantes a insertar: ");
        scanf("%d",&N);
    }while(N<=0);
    for(i=1;i<=N;i++){
        do{
            printf("Edad: ");
            scanf("%d",&edad);
        }while(edad<=0);
        do{
            printf("Indice: ");
            scanf("%f",&indice);
        }while(indice<0||indice>1);
        edadmedia=edadmedia+edad;
        if(indice>0.5){
            edad_regulares=edad_regulares+edad;
            num_regulares=num_regulares+1;
        }
        if(indice>0.9){
            edad_buenr=edad_buenr+edad;
            num_buenr=num_buenr+1;
        }
    }
    printf("\nEdad media de alumnos ingresados: %f",edadmedia/N);
    printf("\nEdad media de alumnos regulares: %f",edad_regulares/num_regulares);
    printf("\nEdad media de alumnos buen rendimiento: %f",edad_buenr/num_buenr);
    return 0;
}
```

Modelo 2:

1. Realice la corrida en frio del siguiente programa e indique su salida.

```

for (i=0;i<N;i++){
    for(j=0;j<N;j++){
        if(j>i)
            printf("* ");
        else
            if(j<i)
                printf("- ");
            else
                printf("+ ");
        printf("\n");
    }
}
    
```

como en el examen anterior pero la salida será para N=4

```

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

2. Escribir un programa que solicite un valor N al usuario y muestre por pantalla todos los números primos menores que N. Se dice que un número es primo cuando solo es divisible por si mismo y el número 1. Por ejemplo: 5 es primo = 1,5 28 no es primo = 1, 2 , 4 , 7 , 14 Realizar Análisis E-P-S y Algoritmo o Codificación.

Igual que el modelo anterior, se calculan los divisores del número. Los números primos solo tienen 2 divisores, el 1 y el número mismo. El ejercicio consiste en contar los divisores del número, si el número de divisores es 2, el número es primo.

- 1 Inicio
- 2 Escribir("Inserte el límite: ")
- 3 Leer(N)
- 4 Para(numero=1;numero<N;numero=numero+1)
  - 4.1 num\_divisores=0 --> cuento el numero de divisores
  - 4.2 Para(i=1;i<numero;i=i+1) -->Para cada numero entre 1 y N, se deben verificar sus divisores
    - 4.2.1 Si (residuo(numero/i)=0) entonces
      - 4.2.1.1 num\_divisores=num\_divisores+1 --> incremento el num de divisores
  - Fin\_si
  - Fin\_RP
  - 4.3 Si(num\_divisores=2) entonces --> si tengo dos divisores lo escribo por pantalla
    - 4.3.1 Escribir(numero) (primo)
  - Fin\_si
  - Fin\_RP
5. Fin

3. Los pacientes con síntomas de una cierta enfermedad son ingresados en el hospital si tienen un valor superior a 0.6 en la medición de un determinado índice, y son operados si el valor es superior a 0.9. Escribir un programa en C que lea desde teclado el número de pacientes a ingresar, seguido de la edad y el índice de cada paciente(valor entre 0 y 1), y calcule la edad media de los pacientes analizados, así como la edad media de los ingresados y la edad media de los operados. Note que un paciente operado también cuenta como ingresado al hospital puesto que su índice también es superior a 0.6 Realizar Análisis E-P-S y algoritmo o codificación. RECUERDE VALIDAR DATOS EN AMBOS EJERCICIOS

Igual que el modelo anterior, pero ahora se trata de un índice de una enfermedad.

```
#include<stdio.h>
int main(){

    int n,i, edad,edad_ing=0,edad_ope=0,num_ing=0,num_ope=0,edad_analizados;
    float indice;
    do{
        printf("\nInserte numero de pacientes: ");
        scanf("%d",&n);
    }while(n<=0);
    for(i=0;i<n;i++){
        do{
            printf(" \nInserte edad ");
            scanf("%d",&edad);
        }while(edad<=0);
        do{
            printf(" \nInserte indice ");
            scanf("%f",&indice);
        }while(indice<0||indice>1);
        edad_analizados=edad_analizados+edad;
        if(indice>0.6){
            edad_ing=edad_ing+edad;
            num_ing=num_ing+1;
        }
        if(indice>0.9){
            edad_ope=edad_ope+edad;
            num_ope=num_ope+1;
        }
    }

    printf("\nPromedios\nAnalizados: %f\nOperados: %f\nIngresados:
%f\n",edad_analizados/n,edad_ope/num_ope,edad_ing/num_ing,edad);
    return 0;
}
```