

Trabajo Práctico Nivel 3

Descomposición de problemas - Modularización (procedimientos)

Fecha de Inicio 08/09/2025 - Fecha de Fin 24/09/2025

Ejercicio 1: Dado el siguiente programa modular, realice el seguimiento de los valores de las direcciones de memoria y de los datos almacenados en las variables a y b, declaradas en el main.

```
1)      # include <stdio.h>
2)      void muestra_parametros(int, int*);
3)      void incrementa_parametros(int, int*);
4)      int main() {
5)          int a,b;
6)          printf("a = "); scanf("%d",&a);
7)          printf("b = "); scanf("%d",&b);
8)          printf("\nDATOS DEL MAIN\n");
9)          printf("valor de a: %d\n",a);
10)         printf("direccion de memoria de a: %p\n",&a);
11)         printf("valor de b: %d\n",b);
12)         printf("direccion de memoria de b: %p\n",&b);
13)         muestra_parametros(a,&b);
14)         incrementa_parametros(a,&b);
15)         muestra_parametros(a,&b);
16)         return 0;
17)     }
18)     void muestra_parametros(int x, int *y) {
19)         printf("\nDATOS DEL MODULO\n");
20)         printf("valor de x: %d\n",x);
21)         printf("direccion de memoria de x: %p\n",&x);
22)         printf("valor de y: %d\n",*y);
23)         printf("direccion de memoria de y: %p\n",y);
24)     }
25)     void incrementa_parametros(int x, int *y) {
26)         x=x+1;
27)         *y=*y+1;
28)     }
```

Ejercicio 2: Diseñe módulos (prototipo más implementación) para cada uno de los casos que se enuncian a continuación, con el propósito de utilizarlos en problemas posteriores:

- Calcule y retorne la cantidad de horas, minutos y segundos que existen en una cantidad de segundos ingresada por el usuario.
- Manipule dos números enteros quitando la última cifra del primero y añadiéndola al final del segundo.
- Reciba dos variables enteras e intercambie sus valores.

Ejercicio 3: Cree un programa que ingrese dos valores enteros A y B y que, a través de un menú, a) muestre el número invertido de A, b) muestre el invertido de B, c) modifique y muestre ambos números, dejando en A el mayor y en B el menor. Utilice, convenientemente, los módulos del ejercicio 2.

Ejercicio 4: Cree un programa que ingrese una fracción de la forma A/B y que a continuación devuelva la fracción irreducible equivalente. ejemplo se $A=6$ y $B=8$; ($6/8$) entonces $Ar=3$ y $Br=4$ ($3/4$) *Nota: Generar un procedimiento que reciba A y B , que devuelva Ar y Br*

Ejercicio 5: Cree un programa que ingrese un número real X , que trabaje con 4 dígitos decimales de precisión, a partir de este número real obtener la parte entera de X y su parte decimal convertida a un número entero.

Nota: Generar un procedimiento que reciba X y devuelva $PentX$, $PdecX$. Ejemplo, si $X=123,5432$, entonces $PentX=123$ y $PdecX=5432$

Ejercicio 6: Dados N números reales, con 4 dígitos decimales de precisión, cree un programa que, para cada número real, cambie cada uno de sus dígitos impares por el dígito par anterior. Ejemplo $A=741,3478$ entonces modificado $A=640,2468$

Nota: reutilizar el modulo generado en el inciso 5

Ejercicio 7: Diseñe un programa que permita el ingreso de N fracciones de la forma (A/B) y muestre la mayor y la menor de ellas, indicando además su expresión irreducible. *Nota: reutilizar el modulo generado en el inciso 4*

Ejercicio 8: Dada una cantidad no determinada de fracciones, de la forma " A/B ", convertir cada fracción en irreducible e indicar si la fracción originalmente ingresada sufrió transformación. *Nota: reutilizar el modulo generado en el inciso 4*

Ejercicio 9: Dada una cantidad no determinada de números naturales, para cada número natural primo, calcular su número invertido y para cada número natural no primo (compuesto) modificarlo eliminando todos sus dígitos primos.

Ejercicio 10: Dada una lista de N fechas ingresadas por el usuario, representadas como números enteros con el formato (AAAADDMM), se pide indicar cuántas fechas son válidas y cuántas no lo son. Se debe tener en cuenta los años bisiestos.

Nota: un año será bisiesto si es divisible entre 4, sin embargo, no puede ser divisible entre 100, a menos que también lo sea por 400. Se debe crear un módulo que separe la fecha en día, mes y año.

Ejercicio 11: Una persona que requiere registrarse en una app, debe elegir una contraseña de 6 dígitos, todos distintos. Si el usuario ingresa un número con cifras repetidas, la app le advierte el error y le sugiere una nueva clave, compuesta por las cifras no repetidas del número ingresado, a la que le agregan nuevos dígitos sin repetir. Los dígitos se agregan siempre como cifra menos significativa. Diseñe e implemente un programa modular, que reciba un número entero y lo modifique de acuerdo al criterio explicado anteriormente.

Ejemplos: clave ingresada = 337498 clave sugerida = 374980 (una posible salida)
clave ingresada = 237744 clave sugerida = 237401 (una posible salida) clave
ingresada = 444444 clave sugerida = 401235 (una posible salida)

Ejercicio 12: Una agencia espacial transmite y recibe mensajes con sus satélites mediante una serie de bits (0 o 1), Como estos mensajes son números grandes, requieren ser codificados para que viajen por el espacio. La codificación consiste en reemplazar grupos de bits de un mismo valor por la cantidad de bits de cada grupo, seguida del valor del bit. Por ejemplo, si el número es = (11000011111) su codificación resulta: (214051). Cuando la agencia recibe un mensaje codificado hace el proceso inverso para descubrir el mensaje original.

Se pide crear un programa para que, mediante un menú de opciones, la agencia pueda enviar o recibir mensajes. Realizar la validación del mensaje a enviar, es decir, que sea efectivamente un número con sólo unos y ceros.

Al realizar la codificación o decodificación correspondientes se debe reemplazar el mensaje según corresponda.

Nota: Asumir que los números binarios a codificar siempre empiezan en 1. Utilizar, de entre los tipos de datos enteros, el que permita el mayor rango para representar el número binario y su codificación. Hacer pruebas con números no muy grandes pues de otra manera la memoria se desbordará. Más adelante se podrán utilizar otros tipos de datos que permitan una mejor solución al problema.