

Trabajo Práctico 2 - Parte 3



Parte 3 - Temas: Funciones predefinidas y definidas por el usuario. Funciones que emplean paso de parámetros por valor y funciones Void. Clases de almacenamiento: static

ES OBLIGATORIO UTILIZAR NOMBRES DE VARIABLES Y DE FUNCIONES SIGNIFICATIVOS

Resolver empleando funciones que retornen o no valor según corresponda. Emplear paso de parámetros por valor en caso de ser necesario

1. Ingresar N caracteres en un programa, identificar cuántos caracteres son numéricos y cuantos alfabéticos. Presentar por pantalla los resultados. Para ello usar funciones para:
 - **int solicitarCantidad()**: solicita el ingreso de N y controla que sea positivo, en caso contrario vuelve a solicitarlo.
 - **int verificarCaracter(char var)**: recibe un carácter e identifica si es alfabético o numérico. Retorna 1 si es numérico y retorna 2 si es alfabético.

2. Ingresar N caracteres en un programa, identificar cuántos caracteres son letras minúsculas e indicar el porcentaje con respecto al total de caracteres. Por ejemplo: se ingresa N= 10 y se ingresan los siguientes caracteres: e 5 T 1 C i ? / 1 >
Se debe mostrar que el porcentaje de minúsculas fue del 20%.

Para ello usar funciones para:

- controlar el valor de N,
- contar cuantos caracteres en minúsculas se ingresaron,
- calcular el promedio con una función con esta declaración:

float calcularPromedio(int cantidad, int total);

- presentar por pantalla el promedio, con una función con la siguiente declaración:
void mostrarPorcentaje(float promedio);

3. Implementar un programa que solicite el ingreso de un valor NUMERO y que muestre por pantalla un mensaje que indique si el valor ingresado es o no un número primo. Usar una función con esta declaración para verificar lo solicitado **int esPrimo(int numero);**
4. Se ingresan **N números enteros**, presentar por pantalla un mensaje indicando cuál es el promedio de los pares. No considerar el número cero como número par. Resolver empleando funciones:
 - Controlar que el valor de N con la función **solicitarCantidad()**.
 - Usar además una función para determinar si un número es par
 - Usar una función para calcular el promedio de solicitado

5. Se ingresan **N números enteros**, presentar por pantalla un mensaje indicando cuál es el promedio de los impares. No considerar el número cero como número impar. Resolver empleando funciones:
 - Controlar que el valor de N con la función **solicitarCantidad()**.
 - Usar además una función para determinar si un número es impar
 - Usar una función para calcular el promedio de solicitado
6. Ingresar **un número entero positivo**, luego presentar por pantalla la cantidad de dígitos que posee dicho número. Para resolver emplear:
 - una función que controle si el dato ingresado es un número positivo, caso contrario solicitar que se ingrese de nuevo.
 - Una función para contar los dígitos del número ingresado
7. Dado un **número natural de 4 o más dígitos**, presentar por pantalla el número y su invertido. Ejemplo : número = 34256, invertido = 65243. Para resolver emplear:
 - Una función para controlar que el número ingresado sea de 4 o más dígitos
 - Una función para invertir el número
8. Codificar un programa que solicite los datos de N usuarios y los almacene en una estructura como el siguiente formato:

```
typedef struct {
    char nombre; //almacenar sólo la inicial
    int dni;
}usuario;
```

Mostrar por pantalla los nombres de los usuarios que fueron ingresados en mayúsculas si su dni es un número par y en minúscula si su dni es un número impar. Por ejemplo se ingresan 3 usuarios:

```
Usuario: a - DNI: 145823
Usuario: t - DNI: 7845152
Usuario: P - DNI: 748452
```

Debe mostrar por pantalla: a , T, P

Para ello usar funciones para:

- **int solicitarCantidad()**: solicita el ingreso de N y controla que sea positivo, en caso contrario volver a solicitarlo.
- Ingresar a cada persona. Esta función debe cumplir con la siguiente declaración: **usuario ingresarUsuario();**
- Mostrar los nombres de cada persona segun fueron ingresados en mayúsculas si su dni es un número par y en minúscula si su dni es un número impar. Esta función debe cumplir con la siguiente declaración: **void presentarUsuario(usuario user);**