



PROGRAMACIÓN

Camino al parcial

Ejercitación

Diseñe e implemente en
Lenguaje C los
siguientes ejercicios



Ejercicios

Se desea crear un programa que genere una versión modificada de la canción infantil en la que se reemplazan todas las vocales por una vocal elegida por el usuario.

La canción original es: "Tengo una hormiguita en la orejita, que me hace mucha cosquillita."

Desarrollar el pseudocódigo que al pedir al usuario que ingrese una vocal (a, e, i, o, u) reemplace todas las vocales de la canción por la vocal ingresada. Mostrar la versión modificada de la canción.

Implementar el algoritmo en lenguaje C

Ejemplo:

Vocal = e

Frase modificada: Tenge ene hermeguate en le erejete, qee me hece meche cesqeellete.

No usar arreglos.

Algoritmo.

ALGORITMO CONVERSION_FRASE

ENTRADA: frase, vocal: carácter.

SALIDA: frase: carácter

**Se considera que la
frase ingresa en
minúsculas**

A1 - HACER

 LEER(vocal)

 MIENTRAS(vocal<>'a' \wedge vocal<>'e' \wedge vocal<>'i' \wedge vocal<>'o' \wedge vocal<>'u')

A2 - LEER(frase)

A3 - MIENTRAS (frase <> '.')

 SI(frase = 'a' \vee frase = 'e' \vee frase = 'i' \vee frase = 'o' \vee frase = 'u')

 frase \leftarrow vocal

 FIN_SI

 ESCRIBIR(frase)

 LEER(frase)

 FIN_MIENTRAS

A4 - PARAR

Código C

```
#include <stdio.h>

int main() {

    char frase, vocal;

    do
    {
        printf("Ingrese una vocal (a, e, i, o, u): ");
        scanf(" %c", &vocal);
    } while (vocal != 'a' && vocal != 'e' && vocal != 'i' && vocal != 'o' && vocal != 'u');

    puts("Ingresar una frase terminada en punto: ");
    scanf("%c", &frase);
    while (frase != '.') {

        if (frase == 'a' || frase == 'e' || frase == 'i' || frase == 'o' || frase == 'u')
        {
            frase = vocal;
        }

        // Ir mostrando la frase modificada
        printf("%c", frase);

        scanf("%c", &frase);

    }

    return 0;
}
```

¿Qué aprendimos?

- ❑ Para control de datos la estructura ideal para usar es do-while.
- ❑ El tratamiento de cadena de caracteres es recomendable utilizar la estructura while() estableciendo una marca final.
- ❑ Al trabajar con cadena de caracteres sin almacenar (sin usar arreglos) se debe analizar, mostrar y leer el próximo caracteres en el mismo bloque de iteración. NO SE PUEDE REALIZAR LAS TAREAS POR SEPARADO.

Ejercicios

Desarrolla un programa que muestre un menú con las siguientes opciones:

- ▣ Calcular el área de un triángulo.
- ▣ Calcular el área de un círculo.
- ▣ Calcular el área de un rectángulo.
- ▣ Salir.

El usuario debe ingresar el número correspondiente a la opción deseada. Luego, el programa deberá solicitar los datos necesarios para realizar el cálculo del área según la opción seleccionada y mostrar el resultado.

Requisitos:

- ▣ Cada opción del menú debe estar implementada en una **función** separada.
- ▣ El programa debe continuar mostrando el menú hasta que el usuario seleccione la opción "Salir".

Algoritmo.

ALGORITMO CALCULO_AREAS

ENTRADA: opcion: entero, base, altura, radio: real

SALIDA: areaCirculo, areaTriangulo, areaRectangulo:real

A1 - HACER

 LEER(opcion)

 SEGUN(opcion)

 1: LEER(base, altura)

 areaTriangulo \leftarrow calcularAreaTriangulo(base, altura)

 ESCRIBIR(areaTriangulo)

 2: LEER(radio)

 areaCirculo \leftarrow calcularAreaCirculo(radio)

 ESCRIBIR(areaCirculo)

 3: LEER(base, altura)

 areaRectangulo \leftarrow calcularAreaRectangulo(base, altura)

 ESCRIBIR(areaRectangulo)

 FIN_SEGUN

 MIENTRAS(opcion < 4)

A2 - PARAR

Algoritmo: Funciones

Función calcularAreaTriangulo(base, altura): real, real -> real

 vble aux: area: real

$area \leftarrow (base * altura) / 2$

 Retorna area

Fin_funcion

Función calcularAreaCirculo(radio): real -> real

 vble aux: area: real

 cte: PI=3,1415

$area \leftarrow PI * radio * radio$

 Retorna area

Fin_funcion

Función calcularAreaCirculo(base, altura): real, real -> real

 vble aux: area: real

$area \leftarrow base * altura$

 Retorna area

Fin_funcion

```

#include <stdio.h>
#define PI 3.1415

float calcularAreaTriangulo(float base, float altura);
float calcularAreaCirculo(float radio);
float calcularAreaRectangulo(float base, float altura);

int main() {
    int opcion; float base, altura, radio;

    do {

        printf("\nMENU DE OPCIONES\n");
        printf("1. Calcular el área de un triángulo\n");
        printf("2. Calcular el área de un círculo\n");
        printf("3. Calcular el área de un rectángulo\n");
        printf("4. Salir\n");
        printf("Seleccione una opción: ");
        scanf("%d", &opcion);

        switch (opcion) {
            case 1: printf("Ingrese la base y altura del triángulo: ");
                    scanf("%f %f", &base, &altura);
                    printf("El área del triángulo es: %.2f\n", calcularAreaTriangulo(base, altura));
                    break;
            case 2: printf("Ingrese el radio del círculo: ");
                    scanf("%f", &radio);
                    printf("El área del círculo es: %.2f\n", calcularAreaCirculo(radio));
                    break;
            case 3: printf("Ingrese la base y altura del rectángulo: ");
                    scanf("%f %f", &base, &altura);
                    printf("El área del rectángulo es: %.2f\n", calcularAreaRectangulo(base, altura));
                    break;
            case 4: printf("Saliendo del programa...\n"); break;
            default: printf("Opción no válida. Intente de nuevo.\n");
        }
    } while (opcion < 4);

    return 0;
}

```

Código C

Código C: funciones

```
float calcularAreaTriangulo(float base, float altura) {  
    return (base * altura) / 2;  
}  
  
float calcularAreaCirculo(float radio) {  
    return PI * radio * radio;  
}  
  
float calcularAreaRectangulo(float base, float altura) {  
    return base * altura;  
}
```

¿Qué aprendimos?



- ❑ La estructura do-while es ideal cuando tenemos un menú de opciones
- ❑ La estructura switch en el lenguaje C solo permite igualdad, por lo tanto para selección de opciones es la estructura de selección perfecta.

Ejercicios

La transposición de una matriz implica intercambiar filas por columnas, es decir, los elementos en la fila i , columna j se convierten en elementos en la fila j , columna i .

Codificar un módulo que dada una matriz $N \times N$ que recibe como argumento, devuelva la transposición de la misma. Realice un programa que invoque a la función y muestre el resultado correspondiente.

Algoritmo.

ALGORITMO MATRIZ TRANSPUESTA

Entrada: matriz: matriz de enteros

Salida: transpuesta : matriz de enteros

vble aux.: i, j: enteros

Cte: tama $\leftarrow 3$

A0 - Inicializar

A1 - transponerMatriz(matriz, transpuesta, tama)

A2 - HACER tama VECES (i=0,..., tama-1)

 HACER tama VECES (j=0,..., tama-1)

 ESCRIBIR(transpuesta_{ij})

 FIN_HACER

FIN_HACER

A3 - PARAR

Algoritmo: Procedimientos

Procedimiento transponerMatriz(matriz, transpuesta, tama):
matriz de enteros, matriz de enteros, entero

Vble AUX: i, j: entero

HACER tama VECES (i=0,..., tama-1)

HACER tama VECES (j=0,..., tama-1)

transpuesta_{ij} \leftarrow matriz_{ji}

FIN_HACER

FIN_HACER

Fin_Procedimiento

```
#include <stdio.h>
#define N 3

void transponerMatriz(int matriz[N][N], int transpuesta[N][N]);
void mostrarMatriz(int matriz[N][N]);

int main() {
    int matriz[N][N] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    int transpuesta[N][N];

    printf("Matriz original:\n");
    mostrarMatriz(matriz);

    transponerMatriz(matriz, transpuesta);

    printf("\n Matriz transpuesta:\n");
    mostrarMatriz(transpuesta);

    return 0;
}
```

Código C

Código C: Funciones

```
void transponerMatriz(int matriz[N][N], int transpuesta[N][N]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            transpuesta[j][i] = matriz[i][j]; // Intercambiar filas por columnas
        }
    }
}

void mostrarMatriz(int matriz[N][N]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d ", matriz[i][j]);
        }
        printf("\n");
    }
}
```

¿Qué aprendimos?

- Si se desea modificar una matriz y hacemos uso de una matriz “auxiliar”, se la debe enviar como parámetro a esa matriz, ya que no se retorna de una función en lenguaje C una matriz. Esto vale arreglos de cualquier dimensión.
- Se deben utilizar tantas estructuras de iteración como dimensiones tenga la matriz.

Ejercicios

Debido a la situación que vive el país, una reconocida financiera desea facilitarles a sus empleados préstamos personales. Se le solicita realizar un sistema que tenga en cuenta los siguientes requisitos:

- El empleado debe tener una antigüedad en la financiera de 5 años como mínimo.
- El sueldo neto debe superar los \$150.000.
- Además debe tener en cuenta el siguiente cálculo de interés al monto de dinero solicitado:
 - Si lo hace a 12 cuotas, al total se aplica un interés del 34%
 - Si lo hace a 24 cuotas, al total se aplica un interés del 48%
 - Si lo hace a 48 cuotas, al total se aplica un interés del 68%

Su tarea será **implementar una función en C** que reciba los datos necesarios y determine el valor de la cuota que deberá pagar el empleado, en caso de corresponder. Realice un programa que invoque a la función y muestre el mensaje correspondiente según el caso: “No apto para solicitar un préstamo” / “La cuota a pagar por el monto elegido es:”.

Asimismo se desea saber cuántas personas aplican para el préstamo, el total de dinero invertido por la financiera y que porcentaje queda fuera de la oportunidad.

Algoritmo.

ALGORITMO PRESTAMO

ENTRADA: numEmpleados, antigüedad, cuotas: entero, sueldoNeto, montoSolicitado: real

SALIDA: totalInvertido, cuota, porcentajeNoAptos: real, empleadosAptos, empleadosNoAptos : entero

A0 - Inicializacion

A1 - LEER(numEmpleados)

A2 - HACER numEmpleados VECES

 LEER(antigüedad, sueldoNeto, montoSolicitado, cuotas)

 cuota \leftarrow calcularCuotaPrestamo(antigüedad, sueldoNeto, montoSolicitado, cuotas)

 SI(cuota \geq 0)

 ESCRIBIR(cuota)

 empleadosAptos \leftarrow empleadosAptos+1

 totalInvertido \leftarrow totalInvertido + montoSolicitado

 SINO

 empleadosNoAptos \leftarrow empleadosNoAptos+1;

 FIN_SI

FIN_HACER

A3 - porcentajeNoAptos \leftarrow (empleadosNoAptos * 100) / numEmpleados;

A4 - ESCRIBIR(empleadosAptos, empleadosNoAptos, porcentajeNoAptos, totalInvertido)

A5 - PARAR

A0 - Inicializacion

 empleadosAptos \leftarrow 0

 empleadosNoAptos \leftarrow 0

 totalInvertido \leftarrow 0

Algoritmo: Función

```
FUNCION calcularCuotaPrestamo(antiguedad, sueldoNeto, montoSolicitado, cuotas): entero, real, real
entero → real
vble Aux: interés, cuota: entero
cuota ← 0
    SI(antiguedad >= 5 && sueldoNeto > 150000)
        SI (cuotas = 12)
            interes ← 0.34
        SINO
            SI (cuotas = 24)
                interes ← 0.48
            SINO
                SI (cuotas = 48) {
                    interes ← 0.68
                SINO
                    cuota←-1
                FIN_SI
            FIN_SI
        FIN_SI
    SI(cuota=0)
        cuota ←(montoSolicitado + (montoSolicitado * interes)) / cuotas
    FIN_SI
SINO
    cuota←-1
FIN_SI

RETORNAR cuota
```

```
#include <stdio.h>
```

```
float calcularCuotaPrestamo(int antigüedad, float sueldoNeto, float montoSolicitado, int cuotas);
```

```
int main() {
```

```
    int numEmpleados, empleadosAptos = 0, empleadosNoAptos = 0, antigüedad, cuotas;  
    float sueldoNeto, cuota, montoSolicitado, totalInvertido=0, porcentajeNoAptos;
```

```
    printf("Ingrese la cantidad de empleados: ");
```

```
    scanf("%d", &numEmpleados);
```

```
    for (int i = 1; i <= numEmpleados; i++) {
```

```
        printf("Empleado %d:\n", i);
```

```
        printf("Antigüedad (en años): ");
```

```
        scanf("%d", &antigüedad);
```

```
        printf("Sueldo neto (en pesos): ");
```

```
        scanf("%f", &sueldoNeto);
```

```
        printf("Monto solicitado (en pesos): ");
```

```
        scanf("%f", &montoSolicitado);
```

```
        printf("Número de cuotas (12, 24, o 48): ");
```

```
        scanf("%d", &cuotas);
```

```
        cuota = calcularCuotaPrestamo(antigüedad, sueldoNeto, montoSolicitado, cuotas);
```

```
        if (cuota > 0) {
```

```
            printf("La cuota a pagar por el monto elegido es: $%.2f\n", cuota);
```

```
            empleadosAptos++;
```

```
            totalInvertido = totalInvertido + montoSolicitado;
```

```
        } else {
```

```
            printf("No apto para solicitar un préstamo.\n");
```

```
            empleadosNoAptos++;
```

```
        }
```

```
    }
```

```
    porcentajeNoAptos = (empleadosNoAptos * 100.0) / numEmpleados;
```

```
    printf("\nResumen:\n");
```

```
    printf("Empleados aptos para el préstamo: %d\n", empleadosAptos);
```

```
    printf("Empleados no aptos para el préstamo: %d\n", empleadosNoAptos);
```

```
    printf("Porcentaje de empleados no aptos: %.2f%%\n", porcentajeNoAptos);
```

```
    printf("Total invertido por la financiera: %.2f\n", totalInvertido);
```

```
    return 0;
```

```
}
```

Código C

Código C: Función

```
float calcularCuotaPrestamo(int antiguedad, float sueldoNeto, float montoSolicitado, int cuotas) {  
    float interes = 0.0, cuota=0;  
  
    if (antiguedad >= 5 && sueldoNeto > 150000)  
    {  
        if (cuotas == 12) {  
            interes = 0.34;  
        } else if (cuotas == 24) {  
            interes = 0.48;  
        } else if (cuotas == 48) {  
            interes = 0.68;  
        } else {  
            cuota = -1.0; // Cuotas no válidas  
        }  
  
        if(cuota ==0)  
        { cuota = (montoSolicitado + (montoSolicitado * interes)) / cuotas; }  
    }else {  
        cuota= -1.0; // No apto para el préstamo  
    }  
  
    return cuota;  
}
```

¿Qué aprendimos?

- Cuando se necesita realizar el cálculo de una operación para un numero de usuarios, la función debe diseñarse para operar sobre una sola persona o usuario. Posteriormente, en el programa principal, se invoca esta función dentro de una estructura de iteración que permita repetir el proceso para cada uno de los usuarios.
- Cuando se piden totales se cuentan dentro de la iteración pero se muestran fuera de la estructura.
- Cuando se solicita porcentajes, se debe realizar el calculo fuera de la estructura de iteración.
- Cuando se solicita que se muestre el calculo realizado para cada usuario, se debe mostrar dentro de la estructura de iteración


```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hasta la próxima clase!!\n");
```

```
    return 0;
```

```
}
```