

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

*Problema 1: Escribir un programa que muestre por pantalla un cuadrado de
dígitos(del 0 al 9) para un valor de n solicitado al usuario.
Si se ingresa un número de dos dígitos, indicar al usuario los valores que
se aceptan.*

VARIABLES DE ENTRADA

*# n (entero): Almacena el número N para dibujar la matriz de números de 1 a
N*

VARIABLES DE SALIDA

*# j (entero): Variable que itera sobre cada fila de la matriz de N x N para
imprimir en pantalla cada uno de sus elementos*

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-1

INICIO

ENTERO i, n, j;

MIENTRAS (1) HACER

IMPRIMIR "Ingrese un valor para n: "

LEER n

SI (n > 0 & n < 9) HACER

PARA i = 1 HASTA i <= n CON PASO 1 HACER

PARA j = 1 HASTA j <= n CON PASO 1 HACER

IMPRIMIR j

FIN_PARA

IMPRIMIR /n

FIN_PARA

BREAK

FIN_SI

DE LO CONTRARIO

IMPRIMIR "El valor de n debe estar en el rango [1,9]"

FIN_DE LO CONTRARIO

FIN

**/*

`#include <stdio.h>`

`int main(){`

`int i, n, j;`

`while(1){`

`printf("Ingrese un valor de n: ");`

`scanf("%i",&n);`

`if(n > 0 && n < 10){`

```

        for(i=1; i<=n; i++){
            for(j=1; j<=n; j++){
                printf("%i: %", j);
            }
            printf("\n");
        }
        break;
    }
    else {
        printf("El valor de n debe estar en el rango de [1,9]");
    }
}
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

2.-Escribir un programa que solicite un número entero N, y que para todo múltiplo X de N menor que 100, calcule el producto de todos los números impares menores que X. El programa deberá mostrar el valor de la suma de todos los productos calculados.

VARIABLES DE ENTRADA

n (entero): Almacena el número N

VARIABLES DE SALIDA

x (entero): Cada uno de los múltiplos de N menores a 100

mult(unsigned long long int): Producto de todos los números impares menores que X

sum(unsigned long long int): Suma de todos los productos mult obtenidos

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-2

INICIO

ENTERO n, x, j=1, i

UNSIGNED LONG LONG INT mult, sum=0

ESCRIBIR "Ingrese el valor de N: "

LEER n

SI n > 0 & n < 100 HACER

*x = j * n*

MIENTRAS x < 100 HACER

mult = 1

PARA i = 1 CON PASO 2 HASTA i < x HACER

*mult = mult * i*

FIN_PARA

```

        ESCRIBIR x, mult
        sum = sum + mult
        j = j + 1
        x = j * n
    FIN_MIENTRAS
    ESCRIBIR "La suma de todos los productos es", sum
FIN_SI
DE LO CONTRARIO HACER
    ESCRIBIR "El valor de N debe ser un número entero entre 1 y 99"
FIN_DE LO CONTRARIO

FIN
*/

#include <stdio.h>
int main(){
    int n, x, j=1, i;
    unsigned long long int mult, sum = 0;
    printf("Ingrese el valor de N: ");
    scanf("%i",&n);
    if(n > 0 & n <100){
        x = j * n;
        while(x < 100){
            mult = 1;
            for(i = 1; i < x; i += 2){
                mult *= i;
            }
            printf("%i: %LLU \n", x, mult);
            sum += mult;
            j++;
            x = j * n;
        }
        printf("Suma de todos los productos: %LLU", sum);
    }
    else {
        printf("El valor de N debe ser un númeromero entero entre 1 y 99 ",
163);
    }
}

```

/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

Problema 3.-Escribir un programa que calcule la función trigonométrica seno en un punto mediante la expresión de un desarrollo en serie de esta.

El valor de x se pedirá al usuario, pero sólo se aceptarán valores comprendidos entre 0 y 20 radianes, considerándose erróneos otros valores. Se considerará

que valor obtenido es correcto cuando el último sumando de la serie anterior sea menor que un error residual máximo (solicitado al usuario). El programa debe mostrar:

- El valor de $\sin(x)$ obtenido utilizando la siguiente instrucción en C: `sin(x)`;
- El valor de $\sin(x)$ calculado haciendo uso del desarrollo en serie anterior.
- El número de iteraciones realizadas para obtener el último valor

VARIABLES DE ENTRADA

x (real): Valor x a evaluar en $\sin(x)$, con $0 \leq x \leq 20$

\maxError (real): Error residual máximo

VARIABLES DE SALIDA

\sin_x (real): La función evaluada en x

i (entero): Total de iteraciones desarrolladas en la serie (debe incrementarse en una unidad)

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-3

FUNCION `fact(entero n)`

 SI $n \geq 1$ HACER

 DEVOLVER $n * \text{fact}(n - 1)$

 FIN_SI

 DE LO CONTRARIO HACER

 DEVOLVER 1

 FIN_DE LO CONTRARIO

FIN_FUNCION

INICIO

 ENTERO $i = 0$, $n = 1$

 REAL x , \maxError

 DOBLE $\sin_x = 0$

 ESCRIBIR "Ingrese el valor de x (en grados): "

 LEER x

$x = x * (3.1459 / 180.0)$

 SI $(x \geq 0) \ \& \ (x \leq 20)$ HACER

 ESCRIBIR "Ingrese el error residual máximo"

 LEER \maxError

 HACER

$\sin_x = \sin_x + (-1 ** i) * (x ** n) / \text{fact}(n)$

$i = i + 1$

```

        n = n + 2
    MIENTRAS maxError <= valorAbsoluto(sin(x) - sin_x)
    ESCRIBIR "Suma de la serie hasta", i + 1, "términos", sin_x
    ESCRIBIR "Sin(", x * (180.0/3.1459), ") usando la biblioteca math.h:
", sin(x)
    FIN_SI
DE LO CONTRARIO HACER
    ESCRIBIR "Por favor, ingrese un valor entre 0 y 20 radianes"
FIN_DE LO CONTRARIO
FIN
*/

```

```

#include <stdio.h>

```

```

#include <math.h>

```

```

int fact(int n){
    if(n >= 1){
        return n*fact(n-1);
    } else {
        return 1;
    }
}

```

```

int main(){
    int i = 0, n = 1;
    float x, maxError;
    double sin_x = 0;
    printf("Ingrese el valor de x (en grados): ");
    scanf("%f", &x);
    // Convirtiendo grados a radianes
    x = x * (3.1459 / 180.0);
    if((x >= 0) & (x <= 20)){
        printf("Ingrese el error residual m%cximo: ", 160);
        scanf("%f", &maxError);
        do {
            sin_x += (pow(-1, i) * (pow(x, n)) / fact(n));
            i++;
            n += 2;
        } while (maxError <= fabs(sin(x) - sin_x));
        printf("\nSuma de la serie hasta %i t%crminos: %Lf \n", (i + 1),
130, sin_x);
        printf("Sin(%.2f) usando la biblioteca math.h: %f \n",
(x*180.0/3.1459), sin(x));
    } else {
        printf("Por favor, ingrese un valor entre 0 y 20 radianes");
    }
}

```

```
}  
}
```

```
/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación  
Nombre del alumno: Francisco Abimael Oro Estrada
```

Problema 4.-Diseñar un algoritmo y realizar su implementación que lea tres números A, B, C y visualice en pantalla el valor del más grande.
Se supone que los tres valores son diferentes

```
## VARIABLES DE ENTRADA
```

```
# nums[3] (Entero): Arreglo que almacena los tres números dados
```

```
## VARIABLES DE SALIDA
```

```
# LargestNumber (Entero): Variable que almacena el número más grande
```

```
## PSEUDOCÓDIGO
```

```
INICIO
```

```
    ENTERO nums[3], LargestNumber, i
```

```
    ESCRIBIR("Ingrese 3 números diferentes: ")
```

```
    PARA i = 0 HASTA i < 3 CON PASO 1 HACER
```

```
        ESCRIBIR "Número", [i], "= "
```

```
        LEER nums[i]
```

```
    FIN_PARA
```

```
    LargestNumber = nums[0]
```

```
    PARA i = 1 HASTA i < 3 CON PASO 1 HACER
```

```
        SI nums[i] > LargestNumber HACER
```

```
            LargestNumber = nums[i]
```

```
        FIN_SI
```

```
    FIN_PARA
```

```
    ESCRIBIR "El número más grande es: ", LargestNumber
```

```
FIN
```

```
*/
```

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

```
int main(){
```

```
    int nums[3], i, LargestNumber;
```

```
    printf("Ingrese tres números diferentes: \n", 163);
```

```
    for(i=0; i<3; i++){
```

```
        printf("Número %i: ", 163, i);
```

```

        scanf("%d", &nums[i]);
    }
    largestNumber = nums[0];
    for(i=1; i<3; i++){
        if(nums[i]>largestNumber){
            largestNumber = nums[i];
        }
    }
    printf("El n%cmero m%cs grande es: %d", 163, 160, largestNumber);
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

5.-Hacer un algoritmo al que le damos la hora HH, MM, SS y nos calcule la hora dentro de un segundo.

Leeremos las horas minutos y segundos como números enteros. Implementar el algoritmo en lenguaje C

VARIABLES DE ENTRADA

inHour[3] (Cadena de caracteres): Arreglo que almacena la hora dada por el usuario.

VARIABLES DE SALIDA

outHour[3] (entero): Arreglo que almacena la hora dentro de un segundo.

PSEUDOCÓDIGO

INICIO

ENTERO outHour[3], i = 0, success = 1

*CHARACTER inHour[8], *token*

ESCRIBIR "Ingrese la hora(HH:MM:SS)"

LEER inHour

token = strtok(inHour, ":")

MIENTRAS(token != NULO) HACER

outHour[i] = atoi(token)

SI (i==0) & (outHour[i] > 23) HACER

ESCRIBIR "Hora no válida"

success = 0

BREAK

FIN_SI

```

        SI (i > 0-) & (outHour[i] < 0 || outHour[i] > 23) HACER
            ESCRIBIR "Hora no válida"
            success = 0
            BREAK
        FIN_SI
        token = strtok(NULL, ":")
        i = i + 1
    FIN_MIENTRAS
    SI success HACER
        outHour[2] = outHour[2] + 1
        PARA i = 2 HASTA i >= 0 CON PASO -1 HACER
            SI outHour[i] > 59 HACER
                outHour[i - 1] = outHour[i - 1] + 1
                outHour[i] = 0
            FIN_SI
        FIN_PARA
        SI outHour[0] > 23 HACER
            outHour[0] = 0
        FIN_SI
        ESCRIBIR "La hora adelantada en un segundo es: "
        PARA i = 0 HASTA i < 3 CON PASO 1 HACER
            SI outHour[i] < 10 HACER
                ESCRIBIR "0"outHour[i]
            FIN_SI
            DE LO CONTRARIO HACER
                ESCRIBIR outHour[i]
            FIN_DE LO CONTRARIO
            SI i < 2 HACER
                ESCRIBIR ":"
            FIN_SI
        FIN_PARA
    FIN_SI
FIN

```

*/

```

#include <string.h>
#include <stdio.h>
#include <stdlib.h>

```

```

int main(){
    int outHour[3], i = 0, success = 1;
    char inHour[8], * token;
    // Leyendo la hora y verificando que sea válida
    printf("Ingrese la hora (HH:MM:SS): ");

```



```

scanf("%s", inHour);
token = strtok(inHour, ":");
while(token != NULL){
    outHour[i] = atoi(token);
    if((i==0) & outHour[i] > 23){
        printf("Hora no v%clida\n", 160);
        success = 0;
        break;
    }
    if((i > 0) & (outHour[i] > 59 || outHour[i] < 0)){
        printf("Hora no v%clida\n", 160);
        success = 0;
        break;
    }
    token = strtok(NULL, ":");
    i++;
}
// Adelantando un segundo a la hora dada
if(success){
    outHour[2]++;
    for(i = 2; i >= 0; i--){
        if(outHour[i] > 59){
            outHour[i-1] += 1;
            outHour[i] = 0;
        }
    }
    if(outHour[0] > 23){
        outHour[0] = 0;
    }
    // Imprimiendo la hora final
    printf("La hora adelantada en un segundo es: ");
    for(i = 0; i < 3; i++){
        if(outHour[i] < 10){
            printf("0%d", outHour[i]);
        }else{
            printf("%d", outHour[i]);
        }
        if(i < 2){
            printf(":");
        }
    }
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

6. Diseñar un programa que permita realizar diferentes tipos de conversiones de monedas.

El usuario debe seleccionar un tipo de conversión desde el menú principal.

Por ejemplo:

- 1. Dólares a pesos*
- 2. Pesos a dólares*
- 3. Pesos a euros*
- 4. Euros a pesos*
- 5. Dólares a euros*
- 6. Euros a dólares*

VARIABLES DE ENTRADA

choice (entero): Almacena el número de la conversión que desea realizar el usuario

inC (real): Almacena la cantidad de la divisa inicial

VARIABLES DE SALIDA

outC (real): Almacena la cantidad convertida a la divisa especificada

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-6

INICIO

ENTERO choice

REAL inC, outC

ESCRIBIR "Opciones:\n 1. Dólares a pesos\n2. Pesos a dólares\n3. Pesos a euros\n4. Euros a pesos\n5. Dólares a euros\n6. Euros a dólares\n", 162, 162, 162, 162

ESCRIBIR "Ingrese la conversión a realizar: ", 162

LEER choice

ESCRIBIR "Ingrese la cantidad inicial: "

LEER inC

SEGUN choice HACER

CASO 1:

*outC = inC * 1.967*

BREAK

CASO 2:

*outC = inC * 19.67 ** (-1)*

BREAK

CASO 3:

*outC = inC * 20.67 ** (-1)*

BREAK

CASO 4:

```

        outC = inC * 20.67
        BREAK
CASO 5:
        outC = inC * 1.05 ** (-1)
        BREAK
CASO 6:
        outC = inC * 1.05
DEFECTO:
        ESCRIBIR "Entrada no v%clida", 160
FIN_SEGUN
        ESCRIBIR "Cantidad final: " outC
FIN
*/
#include <stdio.h>
#include <math.h>
int main(void)
{
    int choice;
    float inC, outC;
    printf("Opciones:\n 1. D%clares a pesos\n2. Pesos a d%clares\n3. Pesos a\n4. Euros a pesos\n5. D%clares a euros\n6. Euros a d%clares\n", 162, 162, 162);
    printf("Ingrese La convers%cn a realizar: ", 162);
    scanf("%i", &choice);
    printf("Ingrese La cantidad inicial: ");
    scanf("%f", &inC);
    switch(choice){
        case 1:
            outC = inC * 19.67;
            break;
        case 2:
            outC = inC * pow(19.67, -1);
            break;
        case 3:
            outC = inC * pow(20.67, -1);
            break;
        case 4:
            outC = inC * 20.67;
            break;
        case 5:
            outC = inC * pow(1.05, -1);
            break;
        case 6:
            outC = inC * 1.05;
            break;
    }
}

```

```

        default:
            printf("Entrada no v%clida", 160);
    }
    printf("Cantidad final: %.2f", outC);
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

7.- Escribir un programa que lea la calificación de un examen por teclado y devuelva la calificación no numérica correspondiente. La calificación podrá ser: No aprobado (0-4.99), Aprobado (5-6.99), Notable (7-8.99), Sobresaliente (9-9.99) o Excelente (10). Realizar este ejercicio utilizando la sentencia de control switch.

VARIABLES DE ENTRADA

fixedGrade (entero): Almacena el valor entero equivalente a la nota ingresada por el usuario

VARIABLES DE SALIDA

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-7

INICIO

ENTERO fixedGrade

ESCRIBIR "Ingrese la calificación: ", 162

LEER fixedGrade

SEGUN fixedGrade HACER

CASO 0 ... 4:

ESCRIBIR "No aprobado"

BREAK

CASO 5 ... 6:

ESCRIBIR "Aprobado"

BREAK

CASO 7 ... 8:

ESCRIBIR "Notable"

BREAK

CASO 9:

ESCRIBIR "Sobresaliente"

BREAK

```

        CASO 10:
            ESCRIBIR "Excelente"
            BREAK
        DEFECTO:
            ESCRIBIR "No es una calificación válida"
    FIN_SEGUN
FIN
*/
#include <stdio.h>

int main(void)
{
    int fixedGrade;
    printf("Ingrese La calificación\n", 162);
    scanf("%d", &fixedGrade);
    switch (fixedGrade){
        case 0 ... 4:
            printf("No aprobado\n");
            break;
        case 5 ... 6:
            printf("Aprobado\n");
            break;
        case 7 ... 8:
            printf("Notable\n");
            break;
        case 9:
            printf("Sobresaliente\n");
            break;
        case 10:
            printf("Excelente\n");
            break;
        default:
            printf("%d no es una calificación válida", fixedGrade, 162,
160);
    }
    return 0;
}

```

/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

8. Repetir el ejercicio anterior, pero utilizando la sentencia de control if-else.

```
## VARIABLES DE ENTRADA
fixedGrade (entero): Almacena el valor entero equivalente a la nota
ingresada por el usuario
```

```
## VARIABLES DE SALIDA
```

```
## PSEUDOCÓDIGO
```

```
ALGORITMO_Oro_Francisco_P2-7
```

```
INICIO
```

```
    ENTERO fixedGrade
```

```
    ESCRIBIR "Ingrese la calificaci%cn\n", 162
```

```
    LEER fixedGrade
```

```
    SI fixedGrade == 10 HACER
```

```
        ESCRIBIR "Excelente"
```

```
    FIN_SI
```

```
    OTRO CASO SI fixedGrade > 9 HACER
```

```
        ESCRIBIR "Sobresaliente"
```

```
    FIN_OTRO CASO
```

```
    OTRO CASO SI fixedGrade > 6 HACER
```

```
        ESCRIBIR "Notable"
```

```
    FIN_OTRO CASO
```

```
    OTRO CASO SI fixedGrade > 4 HACER
```

```
        ESCRIBIR "Aprobado"
```

```
    FIN_OTRO CASO
```

```
    OTRO CASO SI fixedGrade >= 0 HACER
```

```
        ESCRIBIR "No aprobado"
```

```
    FIN_OTRO CASO
```

```
    DE LO CONTRARIO HACER
```

```
        ESCRIBIR "No es una calificación válida"
```

```
    FIN_DE LO CONTRARIO
```

```
FIN
```

```
*/
```

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

```
int main(void)
```

```
{
```

```
    int fixedGrade;
```

```
    printf("Ingrese la calificaci%cn\n", 162);
```

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

```
    if(fixedGrade == 10){
```

```
        printf("Excelente\n");
```

```
    } else if (fixedGrade > 9) {
```

```
        printf("Sobresaliente\n");
```

```
    } else if (fixedGrade > 6){
```

```

        printf("Notable\n");
    } else if (fixedGrade > 4){
        printf("Aprobado\n");
    } else if (fixedGrade >= 0){
        printf("No aprobado\n");
    } else {
        printf("%d no es una calificaci%cn v%clida", fixedGrade, 162,
160);
    }
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

*9. Mejorar el ejercicio anterior de modo que si el usuario introduce un
valor menor que cero o un valor
mayor que 10 se muestre por pantalla un mensaje de error.*

VARIABLES DE ENTRADA

*fixedGrade (entero): Almacena el valor entero equivalente a la nota
ingresada por el usuario*

VARIABLES DE SALIDA

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-7

INICIO

ENTERO fixedGrade, isOver = 0

ESCRIBIR "Ingrese la calificaci%cn\n", 162

LEER fixedGrade

SI fixedGrade < 0 || fixedGrade > 10 HACER

ESCRIBIR "No es una calificación válida"

isOver = 1

SI !isOver HACER

SI fixedGrade == 10 HACER

ESCRIBIR "Excelente"

FIN_SI

OTRO CASO SI fixedGrade > 9 HACER

ESCRIBIR "Sobresaliente"

FIN_OTRO CASO

OTRO CASO SI fixedGrade > 6 HACER

```

        ESCRIBIR "Notable"
    FIN_OTRO CASO
    OTRO CASO SI fixedGrade > 4 HACER
        ESCRIBIR "Aprobado"
    FIN_OTRO CASO
    OTRO CASO SI fixedGrade >= 0 HACER
        ESCRIBIR "No aprobado"
    FIN_OTRO CASO
    FIN_SI
FIN
*/
#include <stdio.h>
int main(void)
{
    int fixedGrade, isOver = 0;
    printf("Ingrese la calificación\n", 162);
    scanf("%d", &fixedGrade);
    if(fixedGrade < 0 || fixedGrade > 10){
        printf("%d no es una calificación válida", fixedGrade, 162,
160);
        isOver = 1;
    }
    if(!isOver){
        if(fixedGrade == 10){
            printf("Excelente\n");
        } else if (fixedGrade > 9) {
            printf("Sobresaliente\n");
        } else if (fixedGrade > 6){
            printf("Notable\n");
        } else if (fixedGrade > 4){
            printf("Aprobado\n");
        } else if (fixedGrade >= 0){
            printf("No aprobado\n");
        }
        return 0;
    }
}

```

/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

10. Escribir un programa que lea cuatro números cualesquiera y determine cuál es el mayor. También deberá

considerar el caso en el que los números sean iguales

VARIABLES DE ENTRADA

nums[4] (real): Arreglo con los números ingresados por el usuario

VARIABLES DE SALIDA

largestNumber (real): Valor más grande del conjunto de números proporcionados por el usuario

PSEUDOCÓDIGO

SubProceso VALUEINARRAY(VAL, ARR[], EXCEPTION)

Para i = 0 Hasta ((variable Reservada sizeofarr)/variable Reservada sizeof(arr[0])) Con Paso 1

Si (exception == i) continue

FinSi

Si (arr[i] == val) return 1

FinSi

FinPara

Algoritmo Oro_Francisco_P2-10

Inicio

entero i

real nums = {0, 0, 0, 0}, largestNumber

Escribir 'Ingrese cuatro números diferentes:', 163

Para i=0 Hasta 4 Con Paso 1

Escribir ': ', i + 1

Leer nums[i]

Si (i == 0)

largestNumber = nums[i]

FinSi

Si (valueinarray(nums[i] nums i))

Escribir 'El número ya se ha ingresado previamente', 163,

nums[i]

i=i-1

FinSi

Si (nums[i] > largestNumber)

largestNumber = nums[i]

FinSi

FinPara

Escribir 'El número ms grande es: ', 163, 160, largestNumber

FinAlgoritmo

*/

#include <stdio.h>

```

int valueinarray(float val, float arr[], int exception)
{
    for(int i = 0; i < sizeof(arr) / sizeof(arr[0]); i++)
    {
        if (exception == i) continue;
        if(arr[i] == val) return 1;
    }
    return 0;
}

int main(void)
{
    int i;
    float nums[4] = {0, 0, 0, 0}, largestNumber;
    printf("Ingresa cuatro n%cmeros diferentes:\n", 163);
    for(i=0; i < 4; i++){
        printf("%i: ", i + 1);
        scanf("%f", &nums[i]);
        if(i == 0){
            largestNumber = nums[i];
        }
        if(valueinarray(nums[i], nums, i)){
            printf("El n%cmero %f ya se ha ingresado previamente\n", 163,
nums[i]);
            i--;
        }
        if(nums[i] > largestNumber){
            largestNumber = nums[i];
        }
    }
    printf("El n%cmero m%cs grande es: %f", 163, 160, largestNumber);
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

10. Escribir un programa que lea cuatro números cualesquiera y determine cuál es el mayor. También deberá considerar el caso en el que los números sean iguales

VARIABLES DE ENTRADA

nums[4] (real): Arreglo con los números ingresados por el usuario

```

## VARIABLES DE SALIDA
largestNumber (real): Valor más grande del conjunto de números
proporcionados por el usuario

## PSEUDOCÓDIGO

SubProceso VALUEINARRAY( VAL, ARR[], EXCEPTION)
    Para i = 0 Hasta ((variable Reservada sizeofarr)/variable Reservada
sizeof(arr[0])) Con Paso 1
        Si (exception == i) continue
        FinSi
        Si (arr[i] == val) return 1
        FinSi
    FinPara
Algoritmo Oro_Francisco_P2-10
Inicio
    entero i
    real nums = {0, 0, 0, 0}, largestNumber
    Escribir 'Ingresa cuatro números diferentes:', 163
    Para i=0 Hasta 4 Con Paso 1
        Escribir ': ', i + 1
        Leer nums[i]
        Si (i == 0)
            largestNumber = nums[i]
        FinSi
        Si (valueinarray(nums[i] nums i))
            Escribir 'El número ya se ha ingresado previamente', 163,
nums[i]
            i=i-1
        FinSi
        Si (nums[i] > largestNumber)
            largestNumber = nums[i]
        FinSi
    FinPara
    Escribir 'El número ms grande es: ', 163, 160, largestNumber
FinAlgoritmo
*/

```

```

#include <stdio.h>

```

```

int valueinarray(float val, float arr[], int exception)
{
    for(int i = 0; i < sizeof(arr) / sizeof(arr[0]); i++)
    {

```

```

        if (exception == i) continue;
        if(arr[i] == val) return 1;
    }
    return 0;
}

int main(void)
{
    int i;
    float nums[4] = {0, 0, 0, 0}, largestNumber;
    printf("Ingresa cuatro n%cmeros diferentes:\n", 163);
    for(i=0; i < 4; i++){
        printf("%i: ", i + 1);
        scanf("%f", &nums[i]);
        if(i == 0){
            largestNumber = nums[i];
        }
        if(valueinarray(nums[i], nums, i)){
            printf("El n%cmero %f ya se ha ingresado previamente\n", 163,
nums[i]);
            i--;
        }
        if(nums[i] > largestNumber){
            largestNumber = nums[i];
        }
    }
    printf("El n%cmero m%cs grande es: %f", 163, 160, largestNumber);
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

11. Escribir un programa que lea cuatro números enteros y determine cuál es el menor. También debe considerar el caso en el que los números sean iguales.

VARIABLES DE ENTRADA

nums[4] (real): Arreglo con los números ingresados por el usuario

VARIABLES DE SALIDA

smallestNumber (real): Valor más pequeño del conjunto de números proporcionados por el usuario

```
## PSEUDOCÓDIGO
```

```
SubProceso VALUEINARRAY( VAL, ARR[], EXCEPTION)  
  Para i = 0 Hasta ((variable Reservada sizeofarr)/variable Reservada  
sizeof(arr[0])) Con Paso 1  
    Si (exception == i) continue  
    FinSi  
    Si (arr[i] == val) return 1  
    FinSi  
  FinPara
```

```
Algoritmo Oro_Francisco_P2-10
```

```
Inicio
```

```
  entero i  
  real nums = {0, 0, 0, 0}, smallestNumber  
  Escribir 'Ingresa cuatro números diferentes:', 163  
  Para i=0 Hasta 4 Con Paso 1  
    Escribir ': ', i + 1  
    Leer nums[i]  
    Si (i == 0)  
      smallestNumber = nums[i]  
    FinSi  
    Si (valueinarray(nums[i] nums i))  
      Escribir 'El número ya se ha ingresado previamente', 163,  
nums[i]  
      i=i-1  
    FinSi  
    Si (nums[i] < smallestNumber)  
      smallestNumber = nums[i]  
    FinSi  
  FinPara  
  Escribir 'El número ms chico es: ', 163, 160, smallestNumber  
FinAlgoritmo  
*/
```

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

```
int valueinarray(float val, float arr[], int exception)  
{  
  for(int i = 0; i < sizeof(arr) / sizeof(arr[0]); i++)  
  {  
    if (exception == i) continue;  
    if(arr[i] == val) return 1;  
  }  
  return 0;  
}
```

```

}

int main(void)
{
    int i;
    float nums[4] = {0, 0, 0, 0}, smallestNumber;
    printf("Ingrese cuatro n%cmeros diferentes:\n", 163);
    for(i=0; i < 4; i++){
        printf("%i: ", i + 1);
        scanf("%f", &nums[i]);
        if(i == 0){
            smallestNumber = nums[i];
        }
        if(valueinarray(nums[i], nums, i)){
            printf("El n%cmero %f ya se ha ingresado previamente\n", 163,
nums[i]);
            --i;
        }
        if(nums[i] < smallestNumber){
            smallestNumber = nums[i];
        }
    }
    printf("El n%cmero m%cs chico es: %.2f", 163, 160, smallestNumber);
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

*12. Escribir un programa que lea cinco números cualesquiera y emita un
mensaje indicando si están o no
ordenados en orden creciente.*

VARIABLES DE ENTRADA

nums[5] (entero): Conjunto de datos proporcionado por el usuario

VARIABLES DE SALIDA

PSEUDOCÓDIGO

Oro_Francisco_P2-12

Algoritmo Oro_Francisco_P2-12

Dimension nums[5]

```

    Escribir 'Ingrese 5 numeros', 163
    Para i = 0 Hasta 5 Con Paso 1
        Escribir ': ', i + 1
        Leer nums[i]
        Si (i == 0) hacer
            prevNumber = nums[i]
        FinSi
        Si (nums[i] < prevNumber) hacer
            isConsecutive = 0
        FinSi
        prevNumber = nums[i]
    FinPara
    Si isConsecutive hacer
        Escribir 'Los nmeros estn ordenados en forma creciente', 163, 160
    FinSi
    De lo contrario hacer
        Escribir 'Los nmeros no estn ordenados en forma creciente', 163, 160
    FinDeLoContrario
FinAlgoritmo
*/

#include <stdio.h>
int main(void)
{
    float nums[5], prevNumber;
    int i, isConsecutive = 1;
    printf("Ingrese 5 n%cmeros\n", 163);
    for (i = 0; i < 5; i++){
        printf("%d: ", i + 1);
        scanf("%f", &nums[i]);
        if(i == 0) prevNumber = nums[i];
        if(nums[i] < prevNumber) isConsecutive = 0;
        prevNumber = nums[i];
    }
    if(isConsecutive) printf("Los n%cmeros est%c\n ordenados en forma
creciente", 163, 160);
    else printf("Los n%cmeros no est%c\n ordenados en forma creciente", 163,
160);
    return 0;
}

```

```

/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

```

13.- Escribir un programa que lea números enteros de teclado hasta que encuentre uno que cumpla las siguientes condiciones:

- Múltiplo de 2.
- No múltiplo de 5.
- Mayor que 100.
- Menor que 10.000.

VARIABLES DE ENTRADA

n(entero): Número ingresado por el usuario

VARIABLES DE SALIDA

n(entero): Número que cumple con las condiciones del problema

PSEUDOCÓDIGO

Algoritmo Oro_Francisco_P2-13

 entero n

 Mientras 1

 Escribir 'Ingrese un numero: ', 163

 Leer n

 Si $(!(n \% 2) \& (n \% 5) \& (n > 100) \& (n < 10000))$

 FinSi

 FinMientras

 Escribir 'El numero cumple con las condiciones del problema', 163, n

FinAlgoritmo

*/

#include <stdio.h>

int main(void)

{

 int n;

 while(1){

 printf("Ingrese un n%cmero: ", 163);

 scanf("%d", &n);

 if $(!(n \% 2) \&\& (n \% 5) \&\& (n > 100) \&\& (n < 10000))$ break;

 }

 printf("El n%cmero %d cumple con las condiciones del problema\n", 163, n);

 return 0;

}

/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

14.- Escribir un programa que diga si un número es primo o no.

VARIABLES DE ENTRADA

n (entero): Número natural

VARIABLES DE SALIDA

PSEUDOCÓDIGO

Algoritmo Oro_Francisco_P2-14

 Escribir 'Ingresa un numero: ', 163

 Leer *n*

 Para *i* = 2 Hasta *n*/ Con Paso 1

 Si $!(n \% i)$

 isPrime = 0

 FinSi

 FinPara

 Si isPrime Escribir 'El numero es primo', 163, *n*

 FinSi

 De lo contrario Escribir 'El numero no es primo', 163, *n*

 FinDeLoContrario

FinAlgoritmo

FIN

*/

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

```
int main(void)
```

```
{
```

```
    int n, i, isPrime = 1;
```

```
    printf("Ingresa un n%cmerno: ", 163);
```

```
    scanf("%d", &n);
```

```
    for (i = 2; i <= n / 2; i++){
```

```
        if  $!(n \% i)$ {
```

```
            isPrime = 0;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if(isPrime) printf("El n%cmerno %d es primo", 163, n);
```

```
    else printf("El n%cmerno %d no es primo", 163, n);
```

```
    return 0;
```

```
}
```

/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación

Nombre del alumno: Francisco Abimael Oro Estrada

15.- Leer 10 valores desde teclado y mostrar la media de los pares y la media de los impares. Hacer tres versiones, con un bucle: for, while y do-while. Repetir el ejercicio considerando que el número de valores se le solicita al usuario.

VARIABLES DE ENTRADA

arr[10] (entero): 10 números enteros proporcionados por el usuario

VARIABLES DE SALIDA

evenMean(real): promedio de los números pares

oddMean (real): promedio de los números impares

PSEUDOCÓDIGO

// Usando for

SubProceso O_E_MEAN_FOR(ARR[10])

Para i=0 Hasta 10 Con Paso 1

Si (arr[i] % 2)

oddSum += arr[i]

oddN=oddN+1

SiNo

evenSum += arr[i]

evenN=evenN+1

FinSi

FinPara

evenMean = evenSum / evenN

oddMean = oddSum / oddN

Escribir 'Media de los números pares: Media de los números impares: ',
163, evenMean, 163, oddMean

FinSubProceso

// Usando while

SubProceso O_E_MEAN_WHILE(ARR[10])

Mientras i < 10

Si (arr[i] % 2)

oddSum += arr[i]

oddN=oddN+1

SiNo

evenSum += arr[i]

evenN=evenN+1

FinSi

i=i+1

FinMientras

```

        evenMean = evenSum / evenN
        oddMean = oddSum / oddN
        Escribir 'Media de Los nmeros pares: Media de Los nmeros impares: ',
163, evenMean, 163, oddMean
FinSubProceso

```

```

// Usando do While
SubProceso O_E_MEAN_DOWHILE( ARR[])
    Hacer
        Si (arr[i] % 2)
            oddSum += arr[i]
            oddN=oddN+1
        SiNo
            evenSum += arr[i]
            evenN=evenN+1
        FinSi
        i=i+1
    Hasta Que (i <= 9)
    evenMean = evenSum / evenN
    oddMean = oddSum / oddN
    Escribir 'Media de Los nmeros pares: Media de Los nmeros impares: ',
163, evenMean, 163, oddMean
FinSubProceso

```

```

Algoritmo Oro_Francisco_P2-15
    Dimension nums[10]
    Escribir 'Ingresa 10 nmeros enteros: ', 163
    Para i = 0 Hasta 10 Con Paso 1
        Escribir ': ', i + 1
        Leer nums[i]
    FinPara
    Escribir 'Usando el for'
    o_e_mean_fornums
    Escribir 'Usando el while'
    o_e_mean_whilenums
    Escribir 'Usando el do while '
    o_e_mean_dowhilenums
FinAlgoritmo
*/

```

```

#include <stdio.h>

```

```

// Usando for
void o_e_mean_for(int arr[10]){
    int i, oddN = 0, evenN = 0, oddSum = 0, evenSum = 0;

```

```

float  evenMean, oddMean;
for(i=0; i < 10; i++){
    if(arr[i] % 2){
        oddSum += arr[i];
        oddN++;
    }
    else{
        evenSum += arr[i];
        evenN++;
    }
}

evenMean = (evenSum / evenN);
oddMean = (oddSum / oddN);
printf("Media de Los n%cmeros pares: %f \nMedia de Los n%cmeros impares:
%f \n", 163,  evenMean, 163, oddMean);
}

```

// Usando while

```

void o_e_mean_while(int arr[10]){
    int i = 0, oddN = 0, evenN = 0, oddSum = 0, evenSum = 0;
    float  evenMean, oddMean;
    while (i < 10)
    {
        if(arr[i] % 2){
            oddSum += arr[i];
            oddN++;
        }
        else{
            evenSum += arr[i];
            evenN++;
        }
        i++;
    }
    evenMean = (evenSum / evenN);
    oddMean = (oddSum / oddN);
    printf("Media de Los n%cmeros pares: %f \nMedia de Los n%cmeros impares:
%f \n", 163,  evenMean, 163, oddMean);
}

```

// Usando do While

```

void o_e_mean_dowhile(int arr[]){
    int i = 0, n = sizeof(arr) / sizeof(arr[0]), oddN = 0, evenN = 0, oddSum
= 0, evenSum = 0;
    float  evenMean, oddMean;

```

```

do
{
    if(arr[i] % 2){
        oddSum += arr[i];
        oddN++;
    }
    else{
        evenSum += arr[i];
        evenN++;
    }
    i++;
} while(i <= 9);
evenMean = (evenSum / evenN);
oddMean = (oddSum / oddN);
printf("Media de Los n%cmeros pares: %f \nMedia de Los n%cmeros impares:
%f \n", 163, evenMean, 163, oddMean);
}

int main(void)
{
    int nums[10], i;
    printf("Ingrese 10 n%cmeros enteros: \n", 163);
    for(i = 0; i < 10; i++){
        printf("%i: ", i + 1);
        scanf("%d", &nums[i]);
    }
    printf("Usando el for\n");
    o_e_mean_for(nums);
    printf("Usando el while\n");
    o_e_mean_while(nums);
    printf("Usando el do while \n");
    o_e_mean_dowhile(nums);
    return 0;
}

```

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación*

Nombre del alumno: Francisco Abimael Oro Estrada

16.- Escribir un programa que pidiendo un valor N seguido de N números, calcule el máximo y mínimo de ese conjunto de N números.

```

## VARIABLES DE ENTRADA
nums[MAXIZE] (entero): Numeros dados por el usuario

## VARIABLES DE SALIDA
largestNumber(entero) Numero mas grande
smallestNumber (entero) Numero mas pequeño

## PSEUDOCÓDIGO
Algoritmo Oro_Francisco_P2-16
Dimension  nums[MAXvariableReservadaZE]
Escribir 'Ingrese el total de nmeros del conjunto: ', 163
Leer n
Para i = 0 Hasta n Con Paso 1
    Escribir ': ', i + 1
    Escribir ', ', i
    Leer nums[i]
    Si (i == 0)
        largestNumber = nums[i]
        smallestNumber = nums[i]
    FinSi
    Si (nums[i] > largestNumber) largestNumber = nums[i]
    Si (nums[i] < smallestNumber) smallestNumber = nums[i]
    FinSi
FinSi
Escribir ' ', largestNumber, smallestNumber
FinAlgoritmo
Escribir 'Valor mximo del conjunto: ', 160, largestNumber
Escribir 'Valor mnimo del conjunto: ', 161, smallestNumber
FinPara

*/

```

```

#include <stdio.h>
#define MAXSIZE 100
int main(void)
{
    int n, nums[MAXSIZE], largestNumber, smallestNumber, i;
    printf("Ingrese el total de n%cmeros del conjunto: ", 163);
    scanf("%d", &n);
    for(i = 0; i < n; i++){
        printf("%d: ", i + 1);
        printf("%d\n", i);
        scanf("%d", &nums[i]);
        if(i == 0){

```

```

        largestNumber = nums[i];
        smallestNumber = nums[i];
    }
    if(nums[i] > largestNumber) largestNumber = nums[i];
    if(nums[i] < smallestNumber) smallestNumber = nums[i];
    printf("%d %d\n", largestNumber, smallestNumber);
}
printf("Valor m%cximo del conjunto: %d\n", 160, largestNumber);
printf("Valor m%cnimo del conjunto: %d\n", 161, smallestNumber);
return 0;
}

```

```

/*
Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

```

17.- Calcula la media de las notas de un conjunto de alumnos. La introducción de datos finaliza cuando el valor de la nota es -1.

```
## VARIABLES DE ENTRADA
```

fixedGrade (entero): Almacena el valor entero equivalente a la nota ingresada por el usuario

```
## VARIABLES DE SALIDA
```

```
*/
```

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

```
int main(void)
```

```
{
```

```
    int n = 0;
```

```
    float sum=0, mean, newGrade;
```

```
    printf("Ingrese la nota %d (-1 para finalizar): ", n + 1);
```

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

```
    sum += newGrade;
```

```
    while(newGrade != -1){
```

```
        n++;
```

```
        printf("Ingrese la nota %d (-1 para finalizar): ", n + 1);
```

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

```

        sum += newGrade;
    }
    mean = sum / n;
    printf("El promedio de las notas es: %.2f ", mean);
}

```

```

/*
Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada

18.- Los empleados de una fábrica trabajan en dos turnos: diurno y nocturno.
Se desea calcular el pago
diario de acuerdo con los siguientes puntos:
1. La tarifa de las horas diurnas es de $5,
2. La tarifa de las horas nocturnas es de $8,
3. caso de ser domingo, la tarifa se incrementará en $2 el turno diurno y $3
el turno nocturno.*/

```

```

#include <stdio.h>
int main(void)
{
    int dhours[2] = {0, 0}, nhours[2] = {0, 0}, total = 0, i, addS;
    printf("Horas diurnas trabajadas: ");
    scanf("%d", &dhours[0]);
    printf("Horas nocturnas trabajadas: ");
    scanf("%d", &nhours[0]);
    printf("¿Agregar horas trabajadas durante el domingo? \t 1 - S%c \t 0 - No\n", 161);
    scanf("%d", &addS);
    if(addS){
        printf("Horas diurnas trabajadas en domingo: ");
        scanf("%d", &dhours[1]);
        printf("Horas nocturnas trabajadas en domingo: ");
        scanf("%d", &nhours[1]);
    }
    total = (dhours[0] * 5 + nhours[0] * 8 + dhours[1] * 7 + nhours[1] * 11);
    printf("Sueldo = %i", total);
}

```


*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

*19.- Implementar el algoritmo para imprimir la suma de los números impares
menores o iguales a N*

VARIABLES DE ENTRADA

*fixedGrade (entero): Almacena el valor entero equivalente a la nota
ingresada por el usuario*

VARIABLES DE SALIDA

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-7

FIN

**/*

#include <stdio.h>

int main(void)

{

int n, sum = 0, i;

printf("Ingrese un número entero: ", 163);

scanf("%d", &n);

for(i = 0; i <= n; i++){

if(i % 2) sum += i;

}

printf("Suma de los números impares menores o iguales a %d: %d", 163,

n, sum);

}

*/*Serie de ejercicios para el segundo parcial de Fundamentos de Programación
Nombre del alumno: Francisco Abimael Oro Estrada*

*20.- Una persona invierte \$1000.00 en una cuenta de ahorro con un 5% de
interés. Se asume que todo
el interés se deja en depósito de la cuenta; calcule y despliegue el monto
acumulado de la cuenta al
final de cada año, durante 10 años. Utilice la siguiente fórmula para
determinar estos montos:*

donde:

p es el monto de la inversión original (inversión principal)

r tasa de interés anual

n número de años

a es el monto del depósito al final del año *n*

VARIABLES DE ENTRADA

fixedGrade (entero): Almacena el valor entero equivalente a la nota ingresada por el usuario

VARIABLES DE SALIDA

PSEUDOCÓDIGO

ALGORITMO_Oro_Francisco_P2-7

FIN

*/

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

```
#include <math.h>
```

```
int main(void)
```

```
{
```

```
    float p = 1000.0, r = 0.05, a;
```

```
    int n = 10, i;
```

```
    for(i = 1; i <= n; i++){
```

```
        a = (p * pow(1 + r, i));
```

```
        printf("Monto del dep%csito al final del a%co %d: %.3f\n", 162, 164,
```

```
i, a);
```

```
    }
```

```
    return 0;
```

```
}
```