

1. Enter this program and compile it. What does it do?

```
#include <stdio.h>
int radius, area;
int main (void){
    printf ("Enter radius ");
    scanf ("%d", &radius);
    area = (int) (3.14159 * radius * radius);
    printf ("\n\nArea = %d\n", area);
    return 0;
}
```

This program calculates the area of a circle. It prompts the user for the radius and then displays the area.

2. Enter this program and compile it. What does it do?

```
#include <stdio.h>
int x, y;
int main (void){
    for (x = 0; x < 10; x++, printf ("\n"))
        for (y = 0; y < 10; y++)
            printf ("X");
    return 0;
}
```

This program prints a 10×10 matrix made of the character X.

3. The following program has a problem. Find it.

```
#include <stdio.h>
int main( void );
{
    printf( "Keep looking!" );
    printf( "You will find it!\n" );
    return 0;
}
```

This program generates a compiler error due to the semicolon at the end of line 3. If you remove the semicolon, this program should compile and run correctly.

4. The following program has a problem. Find it.

```
#include <stdio.h>
int main( void )
{
    printf( "This is a program with a " );
    do_it( "problem!");
    return 0;
}
```

This program generates a compiler error due to the fact that `do_it` is unknown. Change it by `printf` or `puts` and it will work.

5. Make the following change to the program in exercise 2. What does it do now?

```
printf( "X" ); → printf( "%c", 88 );
```

Exactly the same. Because of the values we are providing to `printf`, it uses 88 as if it was a character. Therefore, it uses the ASCII character assigned to 88, that is, "X".

6. Enter this program and compile it. What does it do?

```
#include <stdio.h>
int main(void)
{
    int ctr;
    for( ctr = 65; ctr < 91; ctr++ )
        printf("%c", ctr );
}
```

```

    return 0;
}

```

This program prints the alphabet in all capital letters. The ascii characters corresponding to numbers 65 to 91.

7. Find the error(s) in the following code:

```
printf( "Frosty said, "Hello!"");
```

Double quotes before Hello are considered the end of the string. If you want to use double quotes inside a string, use `\`. You could also use single quote instead of double quotes.

8. Write a program that inputs two floating-point values from the keyboard and displays their product.

```
#include <stdio.h>
float x, y;
int main (void)
{
    puts ("Enter two values: ");
    scanf ("%f %f", &x, &y);
    printf ("\nThe product is %f\n", x * y);
    return 0;
}

```

9. Write a program that prompts for one integer value and then one floating-point value from the keyboard and displays their product.

¿What happens if the user inputs the floating-point value first?

Version 1.

```
#include <stdio.h>
int x;
float y;
int main (void)
{
    puts ("Enter two values: ");
    scanf ("%d", &x);
    scanf ("%f", &y);
    printf ("\nThe product is %f\n", x * y);
    return 0;
}

```

Version 2.

```
#include <stdio.h>
int x;
float y;
int main (void)
{
    puts ("Enter an integer value: ");
    scanf ("%d", &x);
    puts ("Enter an real value: ");
    scanf ("%f", &y);
    printf ("\nThe product is %f\n", x * y);
    return 0;
}

```

In both cases, if you try to introduce a float number first, then:

- The integer part will be considered as x
- The fractional part will be considered as y

!!! BE CAREFUL !!!

10. Write a program that prompts the user for an integer number and outputs that value raised to the power of three.

¿What happens if the user inputs a non integer value?

```
#include <stdio.h>
int a;
int main (void)
{
    puts ("Enter an integer value: ");
    scanf ("%d", &a);
    printf ("\n%d raised to the power of 3 is %d\n", a, a*a*a);
    return 0;
}
```

If the user inputs non integer values, the result will be wrong:

- If it is a non integer number, it is truncated to its integer part (2.5 → 2)
- If it is a non-numeric value, it is converted to 0.

!!! BE CAREFUL !!!

11. Write a program that prompts the user for two values, store them in variables and, after that, swap their values.

```
#include <stdio.h>
int main()
{
    int otra, v1, v2;
    printf( "\n Input v1: " );
    scanf( "%d", &v1 );
    printf( "\n Input v2: " );
    scanf( "%d", &v2 );

    printf( "\n Swapping values..." );
    otra = v1; //v1 is temporary stored in an auxiliar variable
    v1 = v2;
    v2 = otra;

    printf( "\n Now, v1 is: %d", v1 );
    printf( "\n Now, v2 is: %d", v2 );

    return 0;
}
```

12. Write a program that prompts the user for an integer number and shows previous and following numbers.

```
#include <stdio.h>
int main()
{
    int yours;
    printf( "\n Input an integer: ");
    scanf( "%d", &yours );
    printf( "\n Previous number is: %d", yours - 1 );
    printf( "\n Following number is: %d", yours + 1 );
    return 0;
}
```

13. Write a program that prompts the user for his initials (one at a time) and shows them in the format X.Y.Z.

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

```

{
    char nom,ape1,ape2;
    printf( "\n Enter your name's initial: ");
    scanf( "%c", &nom );
    getchar();//avoid "enter" to be "caught" by next "scanf"
    printf( "\n Enter your first surname's initial: ");
    scanf( "%c", &ape1 );
    getchar();
    printf( "\n Enter your second surname's initial: ");
    scanf( "%c", &ape2 );
    printf( "\n Hi, %c.%c.%c!", nom, ape1, ape2 );
    return 0;
}

```

Another option:

```

#include <stdio.h>
int main()
{
    char nom,ape1,ape2;
    printf( "\n Enter your name's initial: ");
    scanf( "\n%c", &nom );
    printf( "\n Enter your first surname's initial: ");
    scanf( "\n%c", &ape1 );
    printf( "\n Enter your second surname's initial: ");
    scanf( "\n%c", &ape2 );
    printf( "\n Hi, %c.%c.%c!", nom, ape1, ape2 );
    return 0;
}

```

14. Write an if statement that assigns the value of x to the variable y only if x is between 1 and 20. Leave y unchanged if x is not in that range.

```

if ((x >= 1) && (x <= 20))
    y = x;

```

15. Use the conditional operator to perform the same task as in exercise 1.

```

y = ((x >= 1) && (x <= 20)) ? x : y;

```

16. Rewrite the following nested if statements using a single if statement and compound operators.

```

if (x < 1)
    if ( x > 10 )
        ...;
if (x < 1 && x > 10 )
    ...;

```

17. What is the result of evaluating the following expressions?

- a. (1 + 2 * 3)
 - b. 10 % 3 * 3 - (1 + 2)
 - c. ((1 + 2) * 3)
 - d. (5 == 5)
 - e. (x = 5)
- 7, 0, 9, 1 (true), 5

18. If x = 4, y = 6, and z = 2, determine whether each of the following evaluates to true or false.

- a. if(x == 4)
- b. if(x != y - z)
- c. if(z = 1)
- d. if(y)

True, false, true, true

19. Write an if statement that determines whether someone is legally an adult (age 18), but not senior yet (age 65).

```
if( age < 18 )
    printf( "You are not an adult" );
else if( age >= 65 )
    printf( "You are senior!");
else
    printf( "You are an adult" );
```

20. What is the value of x when the following statement is complete?

```
for (x = 0; x < 100; x++) ;
100
```

21. What is the value of ctr when the following statement is complete?

```
for (ctr = 2; ctr < 10; ctr += 3) ;
11
```

22. How many Xs does the following print?

```
for (x = 0; x < 10; x++)
    for (y = 5; y > 0; y--)
        puts("X");
50
```

23. Write a for loop to display from 1 to 100 by 3 on screen, starting from 1.

```
int x;
for (x = 1; x <= 100; x += 3)
    printf ("%d\n", x);
```

24. Write a while oop to display from 1 to 100 by 3 on screen, starting from 1.

```
int x=1;
while(x <= 100){
    printf ("%d\n", x);
    x += 3;
}
```

25. Write a do...while oop to display from 1 to 100 by 3 on screen, starting from 1.

```
int x=1;
do{
    printf ("%d\n", x);
    x += 3;
} while(x <= 100);
```

26. What is wrong with the following code fragment?

```
int record = 0;
while (record < 100)
{
    printf( "\nRecord %d ", record );
    printf( "\nGetting next number..." );
}
```

It never ends. It is an infinite loop because record never changes, it is always 0.

27. What is wrong with the following code fragment?

```
#define MAXVAL 50
for (counter = 1; counter < MAXVAL; counter++);
    printf("\nCounter = %d", counter );
```

The semicolon (;) at the end of the for statement.

28. Is anything wrong with the following code?

```
switch( answer )
{
    case `Y`: printf("You answered yes");
               break;
    case `N`: printf( "You answered no");
}

```

NO. Last block within a switch doesn't need a break.

29. Is anything wrong with the following code?

```
switch( choice )
{
    default:
        printf("You did not choose 1 or 2");
    case 1:
        printf("You answered 1");
        break;
    case 2:
        printf( "You answered 2");
        break;
}

```

YES. Not initially, because the default block doesn't need to be at the end. But, if it is not the last block, it does need a break statement.

30. Write a program that provides a menu with three different options. The third option should quit the program. Each of the other options should execute a system command using the system() function.

```
#include <stdio.h>
#include <stdlib.h>
main (){
    char option;
    while(1){//lo que hay dentro se ejecutará indefinidamente
        do{
            printf ("\n\Elige una opcion:\n");
            printf ("a) Crea un archivo vacio llamado \'vacio.txt\'\n");
            printf ("b) Muestra un listado detallado del directorio\n");
            printf ("c) Salir\n\n");
            scanf ("%c", &option);
            getchar();
        } while (option != 'a' && option != 'b' && option != 'c');
        printf("Has elegido la opcion %c\n",option);
        switch (option){
            case 'a':
                system("touch vacio.txt");
                break;
            case 'b':
                system("ls -l");
                break;
            case 'c':
                exit(0);
                break;
            //No hace falta default porque ya hemos controlado antes que sólo acepte a-b-c
        }
    }
}

```

Otra opción:

```
#include <stdio.h>
#include <stdlib.h>
main (){
    char option;
    while(1){//lo que hay dentro se ejecutará indefinidamente
        printf ("\n\Elige una opcion:\n");
        printf ("a) Crea un archivo vacio llamado \'vacio.txt\'\n");
        printf ("b) Muestra un listado detallado del directorio\n");
        printf ("c) Salir\n\n");
        scanf ("%c", &option);
        getchar();
        printf("Has elegido la opcion %c\n",option);
        switch (option){
            case 'a':
                system("touch vacio.txt");
                break;
            case 'b':

```

```

        system("ls -l");
        break;
    case 'c':
        exit(0);
        break;
    default:
        printf("Esa no es una opcion correcta");
    }
}
}

```

31. Escribe un programa que te pida tu nota en números [0-10] y te muestre tu calificación con letras: D [0,5), C [5,7), B [7,9), A [9,10]. Si se introduce un valor no válido, debe seguir preguntando.

```

#include <stdio.h>
main (){
    int nota;
    char calif;
    while(1){//lo que hay dentro se ejecutará indefinidamente
        printf ("\n\nDime tu nota:\n");
        scanf ("%d", &nota);
        getchar()
        switch (nota){
            case 0:
            case 1:
            case 2:
            case 3:
            case 4:
                printf ("\nTu calificación es --> D\n");
                break;
            case 5:
            case 6:
                printf ("\nTu calificación es --> C\n");
                break;
            case 7:
            case 8:
                printf ("\nTu calificación es --> B\n");
                break;
            case 9:
            case 10:
                printf ("\nTu calificación es --> A\n");
                break;
            default:
                printf ("\nNota no válida\n");
        }
    }
}

```

Version 2:

```

#include <stdio.h>
main (){
    int nota;
    char calif;
    while(1){//lo que hay dentro se ejecutará indefinidamente
        printf ("\n\nDime tu nota:\n");
        scanf ("%d", &nota);
        getchar();
        if(nota>=9 && nota <=10){
            printf ("\nTu calificación es --> A\n");
        }else if(nota>=7 && nota <9){
            printf ("\nTu calificación es --> B\n");
        }else if(nota>=5 && nota <7){
            printf ("\nTu calificación es --> C\n");
        }else if(nota>=0 && nota <5){
            printf ("\nTu calificación es --> D\n");
        }else{
            printf ("\nNota no válida\n");
        }
    }
}

```

EXTRA

1. Pide un carácter por teclado y comprueba si es una vocal mayúscula o no.

```
#include <stdio.h>
main (){
    char a;
    printf ("\nEscribe una letra:\n");
    scanf ("%c", &a);
    if(a=='A' || a=='I' || a=='U' || a=='O' || a=='E')
        printf("Sí es una vocal mayúscula");
    else
        printf("NO es una vocal mayúscula");
}
```

2. Pide un carácter por teclado y comprueba si está en mayúsculas.

```
#include <stdio.h>
main (){
    char a;
    printf ("\nEscribe una letra:\n");
    scanf ("%c", &a);
    if(a>=65 && a<=90)
        printf("Sí es una letra mayúscula");
    else
        printf("NO es una letra mayúscula");
}
```

3. Pide un número por teclado y comprueba si es múltiplo de 5.

```
#include <stdio.h>
main (){
    int a;
    printf ("\nDame un número entero:\n");
    scanf ("%d", &a);
    if(a%5==0)
        printf("Sí es un múltiplo de 5");
    else
        printf("NO es un múltiplo de 5");
}
```

4. Pide un año por teclado y comprueba si es bisiesto o no. (Bisiesto: múltiplo de 4, excepto los que son múltiplos de 100 pero no de 400).

```
#include <stdio.h>
main (){
    int a;
    printf ("\nDame un año:\n");
    scanf ("%d", &a);
    if(a%4==0 && !(a%100==0 && a%400!=0))
        printf("Sí es bisiesto");
    else
        printf("NO es bisiesto");
}
```

6. Pide tres números por el teclado y muestra el mayor de ellos.

```
#include <stdio.h>
main (){
    int a,b,c;
    int max;
    printf ("\nDame un entero:\n");
    scanf ("%d", &a);
    printf ("\nDame otro:\n");
    scanf ("%d", &b);
    printf ("\nDame otro mas:\n");
    scanf ("%d", &c);
    max=a;
    if(b>max) max=b;
    if(c>max) max=c;
    printf("el mayor es %d",max);
}
```

Otra opción, con operador condicional (ternario):

```
#include <stdio.h>
main (){
    int a,b,c;
    int max1,max2;
    printf ("\nDame un entero:\n");
    scanf ("%d", &a);
    printf ("\nDame otro:\n");
    scanf ("%d", &b);
    printf ("\nDame otro mas:\n");
    scanf ("%d", &c);
    max1=(b>a)?b:a;
    max2=(c>max1)?c:max1;
    printf("el mayor es %d",max2);
}
```



```
}
```

Otra más:

```
#include <stdio.h>
main (){
    int a,b,c;
    int max;
    printf ("\nDame un entero:\n");
    scanf ("%d", &a);
    printf ("\nDame otro:\n");
    scanf ("%d", &b);
    max=(b>a)?b:a;
    printf ("\nDame otro mas:\n");
    scanf ("%d", &c);
    max=(c>max)?c:max;
    printf("el mayor es %d",max);
}
```

7. Escribe un programa que te permita sumar tantos números como quieras. La cantidad de números a sumar la decide el usuario en cada ejecución. Utiliza un bucle for.

```
#include <stdio.h>
main (){
    int i,cantidad,numero,suma;
    printf ("\nCuantos numeros quieres sumar:\n");
    scanf ("%d", &cantidad);
    suma=0;
    for(i=1;i<=cantidad;i++){
        printf ("\nDame un sumando:\n");
        scanf ("%d",&numero);
        suma+=numero;
    }
    printf("\nLa suma total es %d",suma);
}
```

8. Idem que 7 pero con un bucle while.

```
#include <stdio.h>
main (){
    int i,cantidad,numero,suma;
    printf ("\nCuantos numeros quieres sumar:\n");
    scanf ("%d", &cantidad);
    suma=0;
    i=1;
    while(i<=cantidad)
    {
        printf ("\nDame un sumando:\n");
        scanf ("%d",&numero);
        suma+=numero;
        i++;
    }
    printf("\nLa suma total es %d",suma);
}
```

9. Idem que 7 pero con un bucle do-while.

```
#include <stdio.h>
main (){
    int i,cantidad,numero,suma;
    printf ("\nCuantos numeros quieres sumar:\n");
    scanf ("%d", &cantidad);
    suma=0;
    i=1;
    do{
        printf ("\nDame un sumando:\n");
        scanf ("%d",&numero);
        suma+=numero;
        i++;
    }while(i<=cantidad);
    printf("\nLa suma total es %d",suma);
}
```

10. Pide número enteros positivos constantemente y muestra el resultado de su suma. Si te dan un negativo, lo ignoras. Para terminar el programa, introduce 0. Utiliza un bucle infinito, continue y break.

```
#include <stdio.h>
main (){
    int numero,suma;
    while(1){
        printf ("\nDame un numero positivo:\n");
        scanf ("%d",&numero);
        if(numero < 0) continue;
        else if(numero == 0) break;
        suma+=numero;
    }
}
```

```

    printf("\nLa suma total es %d",suma);
}

```

11. Escribe un programa que muestre por pantalla diez líneas con los 15 primeros múltiplos de los 10 primeros números (1...10). En cada línea deberá mostrar los múltiplos enteros de ese número de línea que no sean múltiplos de 5; los múltiplos de 5 se mostrarán como --. Si en alguna línea se alcanza o supera el valor 50, se dará por terminada la línea.

```

#include <stdio.h>
main (){
    int linea,contador,numero;
    for(linea=1;linea<=10;linea++){
        printf("\nLinea %d:",linea);
        //Lo que viene a continuacion es una linea
        for(contador=1;contador<=15;contador++){
            //Lo que viene a continuacion es un numero dentro de la linea
            numero=linea*contador;
            if(numero>=50) break;//Se termina esta línea, es decir eeste bucle (el otro sigue)
            if(numero%5==0) printf(" --");
            else printf(" %d",numero);
        }
    }
}

```

12. Pide (y exige que sea así) un número entero estrictamente positivo. El programa mostrará cuántas cifras tiene (ej: 3120 → 4).

```

#include <stdio.h>
main (){
    int cifras,original,numero;
    do{
        printf("\nDame un numero: ");
        scanf("%d",&original);
    } while(original<=0);
    cifras=0;
    numero=original;
    do{
        numero=numero/10;
        cifras++;
        //numero=numero%10;
    }while(numero>0);

    printf("\nEl numero %d tiene %d cifras",original,cifras);
}

```

13. Muestra una lista de los factoriales de los X primeros números. X lo elige el usuario en cada ejecución y ha de ser un número entero estrictamente positivo.

Nota: factorial de un número n es $n!=n*(n-1)*(n-2)*...*1$.

Ejemplo: $6!=6*5*4*3*2*1=720$

```

#include <stdio.h>
main (){
    int linea,contador,numero,valor;
    do{
        printf("\nDame un numero: ");
        scanf("%d",&numero);
    } while(numero<=0);
    printf("\nEsta es la lista de los factoriales de los enteros desde 1 hasta %d:",numero);
    for(linea=1;linea<=numero;linea++){
        valor=1;
        for(contador=linea;contador>=1;contador--){
            valor*=contador;
        }
        printf("\n%d-->: %d",linea,valor);
    }
}

```

14. Escribe un programa que lea un número entero positivo menor que 100 e indique si es par o impar.

```

#include<stdio.h>
void main(){
    int numero;
    printf ("Introducir un numero entre 0 y 100: \n");
    scanf ("%d",&numero);
    while (numero>100 || numero<0)
    {
        printf("Error: el numero debe estar enter 0 y 100 \n Introducir un nuevo numero: ");
        scanf ("%d",&numero);
    }
    if (numero%2==0)
        printf("%d es par",numero);
    else
        printf("%d es impar",numero);
}

```

```
}
```

15. Escribe un programa que permita visualizar en pantalla los divisores de un número dado.

```
# include<stdio.h>
void main(){
    int numero,contador;
    printf("Introducir un numero: ");
    scanf("%d",&numero);
    contador=2;
    printf("\nLos divisores de de %d son:",numero);
    while (contador<=numero/2)
    {
        if (numero%contador==0)
            printf("\n%d",contador);
        contador=contador+1;
    }
}
```