

I. ¿Cuál es la salida de estos fragmentos de código?

a. `int a = 5;
int b = 2;
printf("M es %5i y H es %2d\n", a * b, a + b);`

b. `char p = 'A';
printf("El codigo de %c es %d - %x\n", p, p, p);`

c. `char m, n;
n = 0x30;
m = n + 10;
printf("%04d - 0x%02x\n", m, n);`

d. `double t = 6.3673;
printf("%.2f - %05.2f\n", t, t);`

e. `char num1 = 492;
unsigned char num2 = 492;
printf("%d - %d\n", num1, num2);`

f. `int n = 12;
int p = n++;
printf("%d %d\n", n, ++p);`

g. `char a = 54;
unsigned char b = -100;
if (a > b) {
 puts("uno");
} else {
 puts("dos");
}`

h. `printf("%d", 666 > 333);`

i. `int p = 9;
printf("%c", (p % 2 == 0 ? 97 : 66));`

j. `int a = 9, b = 6, c = 3;
a = ++b;
c--;
b = --c - a++;
printf("%d", a + b + c);`

```

k. char opc = '2';
   switch(opc) {
       default:
           printf("N");
       case 1:
           printf("A");
           break;
       case 2:
           printf("B");
           break;
   }

l. int a = 4, b;
   if (a < 2)
       b = 0;
       a = 2;
   if ((a >= 2) && (a <= 6))
       b = 1;
       a = 3;
   if (a == 0)
       b = 5;
   printf("%d", a * b);

m. int p = 200;
   int q = 100;
   int r = 050;
   printf("%d", p * q + r);

n. unsigned char p = 0xA9;
   printf("%x", p ^ 11);
   printf("%x", p | 'A');
   printf("%x", p & 0b00001111);

o. unsigned char p = 200;
   printf("%x", 1 << 5);
   printf("%x", p >> 3);
   p >>= 1;
   printf("%x", p);

p. unsigned char u = 135;
   (u & 0x01) ? printf("impar") : printf("par");

q. int a = 200;
   int b = a > 200 ? 100 : 300;
   printf("%d", b);

r. unsigned char a = 5;
   unsigned char b = 3;
   printf("%0*.*f", a + b + 1, b, 43.1283);

s. unsigned char z = 16;
   z = z << 2;
   printf("%d", z);

```

II. Loops

```
a. int a = 1, s = 0;
   while(a <= 10) {
       s += 2;
       a++;
   }
```

¿Cuánto vale s al salir del loop?

```
b. int a = 0;
   do {
       printf("uno");
       a++;
   } while(a < 0);
```

¿Cuántas veces se muestra "uno" en pantalla?

```
c. unsigned char num = 0;
   while(num > 0) {
       printf("uno");
       num++;
   }
```

¿Cuántas veces se muestra "uno" en pantalla?

```
d. int i = 2;
   for(i = 0; i < 312; i++);
```

¿Cuánto vale i al salir del loop?

```
e. int i = 2;
   for(i = 0; i < 15; i++) {
       i++;
   }
```

¿Cuánto vale i al salir del loop?

¿Qué muestran en pantalla los siguientes fragmentos de código?

```
f. int i;
   for(i = 0; i < 8; i++) {
       if (i % 3 == 0)
           continue;
       else
           printf("%d", i);
   }
```

g. Realizar el ejercicio anterior reemplazando **continue** por **break**

```

h. int sum = 0, i = 1;
   int b[] = {10, 20, 30, 40};
   while (i < 3) {
       sum += b[i - 1];
       i++;
   };
   printf("%d", sum);

i. int sum = 0, i, b[] = {10, 20, 30, 40};
   for(i = 1; i < 3; i++)
       sum *= b[i - 1];
   printf("%d", sum);

j. int e = 1, i;
   for(i = 0; i < 6; i++) {
       switch(e) {
           case 0:
               printf("0");
               e = 1;
               break;
           case 1:
               printf("1");
               e = 0;
               break;
       }
   }

k. int e = 1, cnt = 0;
   for(i = 0; i < 16; i++) {
       cnt++;
       switch(e) {
           case 0:
               printf("0");
               if (cnt == 3) {
                   e = 1;
                   cnt = 0;
               }
               break;
           case 1:
               printf("1");
               if (cnt == 2) {
                   e = 0;
                   cnt = 0;
               }
               break;
       }
   }
}

```

```
1. int e = 7;
   while(e > 5) {
       switch(e) {
           case 7:
               printf("A ");
               e--;
               continue;
               printf("B ");

           default:
               printf("C ");

           case 9:
               printf("D ");
               break;
       }
       e--;
   }
```

III. Funciones

¿Qué muestran en pantalla los siguientes fragmentos de código?

a.

```
int sumar (int a, int b) {  
    int c = a + b;  
    return c;  
}  
  
int main () {  
    int p;  
    p = sumar (sumar (9, 3), 3);  
    printf ("%d ", p);  
}
```

b.

```
int a = 8;  
int sumar(int b) {  
    return a + b;  
}  
  
int main() {  
    int a = 10;  
    int p = sumar(3);  
    printf("%d", p);  
}
```

c.

```
float mayor(float p, float q) {  
    return (p > q) ? p : q;  
}  
  
int main() {  
    printf("%f", mayor(mayor(8, 3), mayor(6, 5)));  
}
```

d.

```
void funcion() {  
    static int p = 2;  
    printf("%d", ++p);  
}  
  
int main() {  
    funcion();  
    funcion();  
}
```

e. Hacer una función que:

Ej. hacer una función que sume tres números enteros

```
#include <stdio.h>

int sumar(int n1, int n2, int n3) {
    int s;
    s = n1 + n2 + n3;
    return s;
}

int main() {
    printf("%d", sumar(1, 4, 6));
}
```

I. Reciba el radio (double) de un círculo y devuelva su superficie.

II. Devuelva 1 si $N > M$, 0 si $N = M$ y -1 si $N < M$ (N y M son enteros y se pasan como parámetros)

III. Reciba un entero N entre 0 y 15 y devuelva un char con la representación de N en hexadecimal.

IV. Punteros

¿Qué muestra el siguiente código?

a.

```
int *q, n = 9;
q = &n;
*q = 7;
printf("%d %d", *q, n);
```

b.

```
int *q, n = 9;
q = &n;
(*q)++;
n = 7;
printf("%d %d", *q, n);
```

c.

```
char v[3] = {1, 3, 4};
char* p = v;
printf("%d %d", *p, *(p + 2));
```

d.

```
char v[5] = {1, 2, 3, 4, 8};
char* p = v;
for(int i = 0; i < 5; i++) {
    printf("%d", *(p + i));
}
```


V. Vectores

- a. Hacer un programa que devuelva el promedio de un vector de enteros (int)

```
#include <stdio.h>

int main() {
    int valores[] = {10, 20, 30, 50, 5, 60, 7, 9, 15, 3, 29, 33, 45, 51, 56 };
    // resolver aquí
}
```

- b. Cree una función que imprima el contenido de un vector de int con el formato
"[0, 1, 2, 3, 4]"

(darle valores al vector al inicializarlo, como en el ejercicio a.)

- c. Cree una función que copie el contenido de un vector en otro:

(darle valores al vector al inicializarlo, como en el ejercicio a.)

- d. Idem utilizando la función memcpy (para usar memcpy hay que agregar string.h)

- e. Hacer una función que reciba tres vectores de enteros p, q y r, y devuelva la cantidad de elementos de p que difieren de q. Al finalizar la función r debe tener contener los elementos de p diferentes de q

Ej.

p[] = [4, 6, 3, 1]

q[] = [4, 1, 3, 6]

la función devuelve 2 y r = [6, 1]

VI. Strings

a. Hacer un programa que muestre las letras de una palabra:

I. en grupos de, como máximo, dos letras:

Ej.

Ingrese una palabra: abalorios

ab
al
or
io
s

II. primero las que están en lugares impares, luego el resto:

Ingrese una palabra: escalera

saea
eclr

b. Hacer un programa que aplique "Cifrado César" a una frase. El usuario ingresa: desplazamiento y dirección, mensaje a cifrar y el programa muestra el alfabeto normal y el desplazado y el mensaje codificado.

Ej.

Ingrese desplazamiento: 6

Ingrese dirección: Derecha

El alfabeto desplazado y codificado:

ABCDEFGHIJKLMNÑOPQRSTUVWXYZ
GHIJKLMNÑOPQRSTUVWXYZABCDEF

Ingrese el texto original: WIKIPEDIA, LA ENCICLOPEDIA LIBRE

El texto codificado es: CÑPÑVKJÑG, QG KSIÑIQUVKJÑG QÑHXX

(ver https://es.wikipedia.org/wiki/Cifrado_C%C3%A9sar)

c. Decodificar un mensaje codificado usando "Cifrado César" El usuario ingresa: desplazamiento, dirección y mensaje a cifrar el mensaje original.

Ej.

Ingrese desplazamiento: 6

Ingrese dirección: Derecha

Ingrese el texto original: CÑPÑVKJÑG, QG KSIÑIQUVKJÑG QÑHXX

El texto codificado es: WIKIPEDIA, LA ENCICLOPEDIA LIBRE

d. Mostrar la abreviatura de una frase ingresada. La abreviatura, para este programa será el conjunto de las primeras letras de cada palabra que forma la frase, en mayúsculas (aunque se haya ingresado en minúsculas) Las palabras están separadas por espacios.

Ingrese frase: Club El Espanto

C.E.E.

- e. Cree una función que devuelva el largo del string (strlen)
<http://www.cplusplus.com/reference/cstring/strlen/?kw=strlen>
- f. Cree una función que copie el string de origen en el vector de destino (strcpy)
<http://www.cplusplus.com/reference/cstring/strcpy/?kw=strcpy>
- g. Implementar una función que haga lo mismo que strncpy.
<http://www.cplusplus.com/reference/cstring/strncpy/?kw=strncpy>
- h. Cree una función que concatene el string de origen en el string de destino. (strcat) <http://www.cplusplus.com/reference/cstring/strcat/?kw=strcat>
- i. Cree una función que busque la primera aparición del string str2 dentro del string str1 (strstr) <http://www.cplusplus.com/reference/cstring/strstr/?kw=strstr>
- j. Desarrolle un programa que muestre la fecha actual con el formato:

aaaa-bb-cc dd:ee:ff

aa: año

bb: mes

cc: día

dd: horas

ee: minutos

ff: segundos

Ejemplo: 2022-09-03 12:45:36

Investigar utilización de función sprintf y biblioteca "time.h".