

Revisar entrega de examen: UF2 2a Convocatoria

| | |
|------------------------|---|
| Usuario | |
| Curso | 2209_ASIR_MP03_Programación básica_ |
| Examen | UF2 2a Convocatoria |
| Iniciado | 4/02/23 19:00 |
| Enviado | 4/02/23 20:02 |
| Fecha de vencimiento | 4/02/23 20:30 |
| Estado | Completado |
| Puntuación del intento | 6,32 de 10 puntos |
| Tiempo transcurrido | 1 hora, 2 minutos |
| Resultados mostrados | Todas las respuestas, Respuestas enviadas, Respuestas correctas, Comentarios, Preguntas respondidas incorrectamente |

Pregunta 1

4 de 4 puntos



Tenemos un programa en C que busca un número introducido por el usuario en un array de números aleatorios con valores generados al azar. El programa muestra cuántas veces se encuentra dicho número dentro del array.

El número a buscar está limitado entre 0 y 10, al igual que los valores generados al azar. El número de elementos del array también se introduce por el usuario, pero está limitado entre 1 y 100.

Todo el código se encuentra dentro de la función principal main(). Transforma el código para que aplique programación modular mediante procedimientos y funciones para:

Pedir un número entre un mínimo y un máximo -> pideNumeroEntreRango

Inicializar el array con el número introducido por el usuario -> inicializarArray

Calcular y mostrar cuántas veces se encuentra el número en el array -> buscarNumero

```
#define MAXARRAY 100
```

```
int main()
```

```
{
```

```
    int numeros[MAXARRAY];
```

```
    int numBuscar;
```

```
    int numE;
```

```
    srand(time(NULL));
```

```
    //INICIALIZACION DEL ARRAY
```

```
    printf("\nIntroduce la cantidad de elementos del array (1-100): ");
```

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

```
    while(numE < 1 || numE > 100)
```

```
    {
```

```
        printf("\nIncorrecto. Introduce un numero entre 1 y 100 ");
```

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

```
    }
```

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

```
    for (int i = 0; i < numE; i++)
```

```
    {
```

```
        numeros[i] = rand() % 10;
```

```
        printf("numeros[%d]=%d\t", i, numeros[i]);
```

```
    }
```

```
    //NUMERO A BUSCAR INTRODUCIDO POR EL USUARIO
```

```
    printf("\nIntroduce el numero que quieres buscar (entre 0 y 10): ");
```

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

```
    while(numE < 0 || numE > 10)
```

```
    {
```

```
        printf("\nIncorrecto. Introduce un numero entre 0 y 10: ");
```

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

```
    }
```

```
    //BUSCAR CUANTAS VECES SE ENCUENTRA EL NUMERO DENTRO DEL ARRAY
```

```
    int contadorNumeros = 0;
```

```
    for (int i = 0; i < numE; i++)
```

```
    {
```

```
        if(numeros[i] == numBuscar)
```

```
        {
```

```
            contadorNumeros++;
```

```
        }
```

```
    }
```

```
    printf("\nEl numero %d se encuentra %d veces en el array", numBuscar, contadorNumeros);
```

```
    getch();
```

```
    return 0;
```

```
}
```

Bloques preparados para las funciones. Puedes copiar estos bloques para usarlos en tu respuesta, rellenandolos con el código, parámetros y valor de retorno que creas que llevaría.

```
_____ pideNumeroEntreRango(_____)
```

```
{
```

```
}
```

```
_____ inicializarArray(_____)
```

```
{
```

```
}
```

```
_____ buscarNumero(_____)
```

```
{
```

```
}
```

Respuesta
seleccionada:

Tenemos un programa en C que busca un número introducido por el usuario en un array de números aleatorios con valores generados al azar. El programa muestra cuántas veces se encuentra dicho número dentro del array. El número a buscar está limitado entre 0 y 10, al igual que los valores generados al azar. El número de elementos del array también se introduce por el usuario, pero está limitado entre 1 y 100. Todo el código se encuentra dentro de la función principal main(). Transforma el código para que aplique programación modular mediante procedimientos y funciones para:
Pedir un número entre un mínimo y un máximo -> pideNumeroEntreRango
Inicializar el array con el número introducido por el usuario -> inicializarArray
Calcular y mostrar cuántas veces se encuentra el número en el array -> buscarNumero

```
#define MAXARRAY 100
int pideNumeroEntreRango(int min, int max);
int inicializarArray(int min, int max);
int buscarNumero()

int main()
{
    int numeros[MAXARRAY];
    int numBuscar;
    int numE;

    srand(time(NULL));

    //INICIALIZACION DEL ARRAY
    printf("\nIntroduce la cantidad de elementos del array (1-100): ");
    scanf("%d", &numE);
    while(numE < 1 || numE > 100)
    {
        printf("\nIncorrecto. Introduce un numero entre 1 y 100 ");
        scanf("%d", &numE);
    }
    printf("\n\n");
    for (int i = 0; i < numE; i++)
    {
        numeros[i] = rand() % 10;
        printf("numeros[%d]=%d\t",i,numeros[i]);
    }

    //NUMERO A BUSCAR INTRODUCIDO POR EL USUARIO
    printf("\nIntroduce el numero que quieres buscar (entre 0 y 10): ");
    scanf("%d", &numBuscar);
    while(numE < 0 || numE > 10)
    {
        printf("\nIncorrecto. Introduce un numero entre 0 y 10: ");
        scanf("%d", &numE);
    }

    //BUSCAR CUANTAS VECES SE ENCUENTRA EL NUMERO DENTRO DEL ARRAY
    int contadorNumeros = 0;
    for (int i = 0; i < numE; i++)
    {
        if(numeros[i] == numBuscar)
        {
            contadorNumeros++;
        }
    }
    printf("\nEl numero %d se encuentra %d veces en el array", numBuscar, contadorNumeros);

    getch();
    return 0;
}

int pideNumEntreRango(int min, int max)
{
    int num;
    printf("\nIntroduce un numero entre %d y %d: ", min, max);
    scanf("%d", &num);
    while( num<min || num>max )
    {
        printf("\nIntroduce un numero entre %d y %d: ", min, max);
        scanf("%d", &num);
    }
    return num;
}

void inicializarArray(int numeros[], int numE){
    for (int i = 0; i < numE; i++){
        numeros[i] = rand() % 10;
        printf("numeros[%d]=%d\t",i,numeros[i]);
    }
    int buscarNumero(int numeros[], int numE, int numBuscar){
        if(numeros[i] == numBuscar) {
            contadorNumeros++;
            return contadorNumeros;
        }
    }
}
```

Respuesta correcta: [None]
Comentarios para
respuesta: [No se ha dado ninguna]



Queremos hacer un programa en C que calcule el número de segundos que hay en una cantidad de horas y minutos indicadas por el usuario. También tiene que hacer el cálculo inverso; pedir al usuario una cantidad de segundos y calcular las horas y minutos.

Para pedir el número de horas, minutos y segundos se usará la función pideNumEntreRango, que pide un valor mínimo y máximo y devuelve un número comprendido entre ambos.

Para calcular la cantidad de segundos dadas las horas y minutos, usará la función calculaSegundos, que recibe la cantidad de horas y minutos y devuelve los segundos.

Para calcular las horas y minutos dados los segundos, usará la función calculaHorasMinutos, que recibe la cantidad de segundos y asigna el valor de horas y minutos a los otros dos parámetros que recibe.

Las horas estarán limitadas entre 0 y 24, los minutos entre 0 y 60, y los segundos entre 0 y 99999.

```
int pideNumEntreRango(int min, int max);

int calculaSegundos(int horas, int minutos);

[1] calculaHorasMinutos( int segundos, [2] horas, [3] minutos);

int main()
{
    //Variables para guardar horas, minutos y segundos

    int horas;
    int minutos;
    int segundos;

    //Pedir numero de horas y minutos
    printf("Numero de horas");

    horas = pideNumEntreRango(0, 24);
    printf("\nNumero de minutos");
    minutos = pideNumEntreRango(0, 60);

    //Calcular el numero de segundos
    segundos = calculaSegundos ( [4] , [5] );
    printf("\n %d horas y %d minutos en segundos son: %d", horas, minutos, segundos);

    //Pedir el numero de segundos para calcular horas y minutos
    printf("\n\nNumero de segundos");
    segundos = pideNumEntreRango(0, 99999);
    //Calcular la cantidad de horas y minutos
    calculaHorasMinutos( segundos , [6] , [7] );

    printf("\n%d segundos equivale a %d horas y %d minutos", segundos, horas, minutos);

    getch();
    return 0;
}

int pideNumEntreRango(int min, int max)
{
    int num;

    printf("\nIntroduce un numero entre %d y %d: ", min, max);
    scanf("%d", &num);

    //Comprueba que el numero introducido esta entre el máximo y el mínimo
    while( [8] || [9] )
    {
        printf("\nIntroduce un numero entre %d y %d: ", min, max);
        scanf("%d", &num);
    }

    return num;
}
```

//Los segundos a partir de los minutos se calculan multiplicando los minutos por 60. A partir de las horas se calculan multiplicando horas por 3600.

int calculaSegundos(int horas, int minutos)

















```
{  
    int minComoSegundos = minutos * 60;  
    int horasComoSegundos = horas * 3600;  
  
    return [10] ;  
}
```
















//Las horas a partir de los segundos se calculan dividiendo los segundos entre 3600.

//Los minutos a partir de los segundos se calculan a partir del resto de la división de los segundos entre 3600, y dividiendo el valor de ese resto entre 60.

[11] calculaHorasMinutos(int segundos, [12] horas, [13] minutos)

```
{  
    [14] = segundos / 3600;  
    [15] = ( [16] % 3600) / 60;  
}
```

Respuesta especificada para: 1  int
 Respuesta especificada para: 2  int
 Respuesta especificada para: 3  int
 Respuesta especificada para: 4  int horas
 Respuesta especificada para: 5  int minutos
 Respuesta especificada para: 6  horas
 Respuesta especificada para: 7  minutos
 Respuesta especificada para: 8  <min
 Respuesta especificada para: 9  >max
 Respuesta especificada para: 10  segundos
 Respuesta especificada para: 11  void
 Respuesta especificada para: 12  int
 Respuesta especificada para: 13  int
 Respuesta especificada para: 14  horas
 Respuesta especificada para: 15  minutos
 Respuesta especificada para: 16  segundos

| | | |
|--|--|---|
| Respuestas correctas para: 1 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | void | |
| Respuestas correctas para: 2 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | int * | |
| Respuestas correctas para: 3 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | int * | |
| Respuestas correctas para: 4 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | horas | |
| Respuestas correctas para: 5 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | minutos | |
| Respuestas correctas para: 6 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | &horas | |
| Respuestas correctas para: 7 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | &minutos | |
| Respuestas correctas para: 8 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | num < min | |
| Respuestas correctas para: 9 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | num > max | |
| Respuestas correctas para: 10 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | minComoSegundos + horasComoSegundos | |
| Respuestas correctas para: 11 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | void | |
| Respuestas correctas para: 12 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | int * | |
| Respuestas correctas para: 13 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | int * | |
| Respuestas correctas para: 14 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | *horas | |
| Respuestas correctas para: 15 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |
|  Correspondencia exacta | *minutos | |
| Respuestas correctas para: 16 | | |
| Método de evaluación | Respuesta correcta | Distingue entre mayúsculas y minúsculas |

Pregunta 3

0,75 de 1 puntos



Explica que diferencia hay entre un procedimiento y una función. Pon un ejemplo práctico en C de cuando usarías un caso u otro.

Respuesta
seleccionada:

Un procedimiento no devuelve ningún valor al main principal, en cambio una función si que devuelve un valor, ya sea entero, float, etc...

```
funcion ==> int calcularNumeros();
```

```
procedimiento ==> void calcularNumeros();
```

Respuesta correcta:

[None]

Comentarios para
respuesta:

Faltaría algo de código de ejemplo en el procedimiento y la función: por ejemplo, la función calcula una suma y devuelve el valor, y el procedimiento solo muestra mensajes por pantalla.

lunes 13 de febrero de 2023 13H45' CET

← Aceptar