APRENDERAPROGRAMAR.COM

EJERCICIOS CON FUNCIONES EN C. EJEMPLO CALCULAR SERIES NUMÉRICAS. REFACTORIZAR. (CU00552F)

Sección: Cursos
Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

Resumen: Entrega nº52 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

EJERCICIO RESUELTO Nº1: ENUNCIADO

Desarrollar un programa con las siguientes opciones:

a) Introducir un valor entero impar comprendido entre 1 y 19
b) Calcular la serie numérica *1 + 3 + 5 + ··· + n*
c) Calcular *1 * 3 * 5 * ··· * n*
d) Salir del programa.

Nota: Los cálculos se han de realizar a través de un array que almacene los valores impares comprendidos entre *1* y n, usando bucles y acumuladores para recorrer el vector y obtener resultados.

EJERCICIO RESUELTO Nº1: SOLUCIÓN

Aunque el problema podría resolverse usando fórmulas como:

$$1+3+5+\cdots+n=\left(\frac{n+1}{2}\right)^2$$

Vamos a resolverlo tal y como indica el enunciado, usando arrays. Aunque a priori no conozcamos el valor entero impar con que hemos de realizar los cálculos, sí sabemos que ha de estar comprendido entre 1 y 19. Esto significa que como mucho tendremos que operar con diez datos (el número de impares que hay entre 1 y 19, es decir, 1, 3, 5, 7, 9, 11, 13, 15, 17 y 19).

```
//Programa cálculos con impares01
#include <stdio.h>
#include <stdlib.h>

#define NMAX 19

int valor[NMAX]; //Declaración de array como variable global, realmente solo usaremos 10 elementos
int numeroItems; //Declaración como variable global
int obtenerValorParaCalculo(); void rellenarArray (int numero);
long valorSumatorio (int numeroItemsSum); long valorProducto (int numeroItemsProd);
// Ejercicios resueltos de programación aprenderaprogramar.com

int main() {
   int valorParaCalculo=0;    int opcionUsuario = 1;
   while (opcionUsuario!=4) {
      printf("\n\n1: Introducir valor\n2: Calcular serie 1+3+5...\n");
      printf("3: Calcular serie 1*3*5...\n4: Salir\n\nElija opcion: ");
      scanf("%d", &opcionUsuario);
```

```
    if (opcionUsuario==1) {
      valorParaCalculo =  obtenerValorParaCalculo();
      rellenarArray (valorParaCalculo);
    }
if (opcionUsuario==2) {
      if (valorParaCalculo!=0) {
        printf("n = %d El valor del sumatorio es: %ld", valorParaCalculo, valorSumatorio(numeroItems));
      } else {
        printf("No hay un valor para calculo que permita obtener sumatorio 1+3+5+...");
      }
    }
    if (opcionUsuario==3) {
      if (valorParaCalculo!=0) {
        printf("n = %d El valor del producto es: %ld", valorParaCalculo, valorProducto(numeroItems));
      } else {
        printf("No hay un valor para calculo que permita obtener resultado 1*3*5*...");
      }
    }
  }
  return 0;
}


int obtenerValorParaCalculo() {
  int imparElegido = 0;
  do {
    printf("\nPor favor introduzca numero entero impar entre 1 y 19: ");
    scanf("%d", &imparElegido);
  } while (imparElegido<=0 || imparElegido> NMAX ||imparElegido%2==0);
  return imparElegido;
}

void rellenarArray (int numeroElegido) {
  int impar = -1;    int i=0;
  do {
    i = i+1;
    impar = impar +2;
    valor[i] = impar;
  } while (impar < numeroElegido);
  numeroItems = i;
}

long valorSumatorio (int numeroItemsSum) {
  long resultadoSum =0;   int i=0;
  for (i=1; i<=numeroItemsSum; i++) {
    resultadoSum = resultadoSum + valor[i];
  }
  return resultadoSum; //Devolvemos el valor del sumatorio
}

long valorProducto (int numeroItemsProd) {
  int i=0;    long resultadoProd = 1;
  for (i=1; i<=numeroItemsProd; i++) {
    resultadoProd = resultadoProd * valor[i];
  }
  return resultadoProd; //Devolvemos el valor del producto
}
```

Comentarios: Distintas variantes pueden ser introducidas para resolver este ejercicio. Te animamos a que intentes resolverlo de otras maneras. Para algunas variables hemos usado el tipo de dato *long* en lugar del tipo *int*. Con ello pretendemos evitar posibles problemas relacionados con las limitaciones que para los rangos numéricos supone trabajar con *int*. Al realizar cálculos con sucesiones geométricas los valores numéricos crecen muy rápidamente y a pesar de que *int* admite valores grandes, pueden ser insuficientes en algunos casos.

Un ejemplo de resultado que podemos obtener con este programa es el siguiente:

```
1: Introducir valor 2: Calcular serie 1+3+5...3: Calcular serie 1*3*5... 4: Salir
Elija opcion: 1          Por favor introduzca numero entero impar entre 1 y 19: 9
Elija opcion: 2          n = 9 El valor del sumatorio es: 25
Elija opcion: 3          n = 9 El valor del producto es: 945
Elija opcion: 4
```

EJERCICIO RESUELTO Nº2: ENUNCIADO

Desarrollar un programa con las siguientes opciones:

e) Introducir un valor entero impar comprendido entre 1 y 19
f) Calcular la serie numérica $1 + 3 + 5 + \cdots + n$
g) Calcular $1 * 3 * 5 * \cdots * n$
h) Salir del programa.

Nota: Los cálculos se han de realizar sin hacer uso de arrays. Este ejercicio es análogo (mismo enunciado) al anterior. Lo que pretendemos con su resolución es explorar distintas formas de resolver un mismo problema, aún sin plantearnos que sean formas de resolución óptimas.

EJERCICIO RESUELTO Nº2: SOLUCIÓN

```
//Programa cálculos con impares02
#include <stdio.h>
#include <stdlib.h>

#define NMAX 19
int obtenerValorParaCalculo(); void resolverSumatorio (int numeroItemsSum);
void resolverProducto (int numeroItemsProd);

int main() {
   int valorParaCalculo=0;   int opcionUsuario = 1;
   while (opcionUsuario!=4) {
      printf("\n\n1: Introducir valor\n2: Calcular serie 1+3+5...\n");
      printf("3: Calcular serie 1*3*5...\n4: Salir\n\nElija opcion: ");
      scanf("%d", &opcionUsuario);
      if (opcionUsuario==1) {
         valorParaCalculo =  obtenerValorParaCalculo();
      }
```

```
if (opcionUsuario==2) {
        if (valorParaCalculo!=0) {
           resolverSumatorio(valorParaCalculo);
        } else {
           printf("No hay un valor para calculo que permita obtener sumatorio 1+3+5+...");
        }
     }
     if (opcionUsuario==3) {
        if (valorParaCalculo!=0) {
           resolverProducto(valorParaCalculo);
        } else {
           printf("No hay un valor para calculo que permita obtener resultado 1*3*5*...");
        }
     }
  }
  return 0;
}


int obtenerValorParaCalculo() {
  int imparElegido = 0;
  do {
     printf("\nPor favor introduzca numero entero impar entre 1 y 19: ");
     scanf("%d", &imparElegido);
  } while (imparElegido<=0 || imparElegido>NMAX ||imparElegido%2==0);
  return imparElegido; // Ejercicios resueltos de programación aprenderaprogramar.com
}

void resolverSumatorio (int valorCalculoSum) {
  int impar = -1;     long resultadoSum =0;
  do {
     impar = impar + 2;
     resultadoSum = resultadoSum + impar;
  } while (impar < valorCalculoSum);
  printf("n = %d El valor del sumatorio es: %ld", valorCalculoSum, resultadoSum);
}

void resolverProducto (int valorCalculoProd) {
  int impar = -1;     long resultadoProd = 1;
  do {
     impar = impar + 2;
     resultadoProd = resultadoProd * impar;
  } while (impar < valorCalculoProd);
  printf("n = %d El valor del producto es: %ld", valorCalculoProd, resultadoProd);
}
```

EJERCICIO

Reescribe el programa anterior para que todas las funciones tengan un tipo de retorno y que los mensajes no sean mostrados por las distintas funciones, sino que la muestra de mensajes sea realizada exclusivamente por la función *main*. Esto se corresponde con reorganizar o reescribir un código para mejorarlo, y a esto se le suele denominar "refactorización del código".

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00553F

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210