

FUNDAMENTOS DE PROGRAMACIÓN
LABORATORIO 7
PROPUESTAS DE SOLUCIÓN
SEMESTRE ACADÉMICO 2024-0

Horarios: Todos los horarios

Elaborado por Mag. Rosa Latorraca

INDICACIONES:

- Debe utilizar variables descriptivas, comentarios y mensajes descriptivos.
- El orden y la eficiencia de su implementación serán considerados en la calificación.

RESULTADOS ESPERADOS:

- Al finalizar la sesión, el alumno comprenderá el funcionamiento de la estructura algorítmica selectiva múltiple.
- Al finalizar la sesión, el alumno construirá programas usando estructuras algorítmicas selectivas múltiple.
- Al finalizar la sesión, el alumno comprenderá el funcionamiento de la estructura algorítmica iterativa con salida controlada.
- Al finalizar la sesión, el alumno construirá programas usando la estructura algorítmica iterativa con salida controlada.

CONSIDERACIONES:

- La solución presentada para cada problema corresponde a una propuesta de solución por parte del autor.
- En programación pueden existir muchas soluciones para un mismo problema pero debe cumplir con todo lo solicitado, incluyendo las restricciones brindadas.

Desarrolle los siguientes problemas en lenguaje C:

1. Números Zuckerman y alternos

Un número es **Zuckerman** si es divisible por el producto de todos sus dígitos.

Por ejemplo:

- El número 224 es un número Zuckerman porque 224 es divisible por $2*2*4$.
- El número 142 no es un número Zuckerman porque 142 no es divisible por $1*4*2$.

Un número es **alterno** si se cumple la siguiente propiedad: si a este número se le alternan dígitos pares e impares.

Por ejemplo:

- El número 492 es un número alterno porque se alternan los dígitos pares e impares. La unidad es par, la decena es impar y la centena es par.
- El número 190 no es un número alterno porque no se alternan los dígitos pares e impares. La unidad es par, la decena es impar pero la centena no es par.

Se pide desarrollar un programa en Lenguaje C que permita realizar lo siguiente:

- Muestre un menú de opciones, las cuales son **Z**: Números Zuckerman. **A**: Números alternos.
- Leer una opción. Debe validar que la opción ingresada sea **Z** o **A**. Si elige una opción distinta a las indicadas, se mostrará el mensaje: `Opción inválida` y el programa debe terminar.
- Leer el rango de números.
- Debe validar que el límite inferior del rango sea mayor a 0, el límite inferior sea menor al límite superior y el límite superior sea menor a 1000. En caso contrario, debe emitir el siguiente mensaje: `El rango ingresado es inválido` y el programa debe terminar.
- Si pasó todas las validaciones, debe mostrar los números Zuckerman o alternos que existen en el rango, según la opción ingresada.

La solución deberá contar como mínimo 3 módulos, incluido el módulo principal. Además, uno de los módulos debe devolver como mínimo dos valores como parámetros que modifican su valor y no puede usarse para leer datos.

Consideraciones adicionales

Para la solución del problema debe tener en cuenta lo siguiente:

- Para el cálculo de la cantidad de cifras del número use la función `logaritmo en base 10 del número` y adicione uno.
- Considerar como opciones válidas solo las letras **Z** y **A** en mayúsculas.

A continuación se presentan los siguientes ejemplos de ejecución:

Casos de Prueba:

```
Menú de opciones:  
Z: Números Zuckerman.  
A: Números alternos.
```

```
Ingrese una opción:  
a
```

```
Opción inválida.
```

```
Menú de opciones:  
Z: Números Zuckerman.  
A: Números alternos.
```

```
Ingrese una opción:  
Z
```

```
Ingrese el rango: -5 10
```

```
El rango ingresado es inválido.
```

```
Menú de opciones:  
Z: Números Zuckerman.  
A: Números alternos.
```

```
Ingrese una opción:  
A
```

Ingrese el rango: 500 1020

El rango ingresado es inválido.

Menú de opciones:

Z: Números Zuckerman.

A: Números alternos.

Ingrese una opción:

Z

Ingrese el rango: 10 35

En el rango [10,35] hay los siguientes números Zuckerman:

11 12 15 24

Menú de opciones:

Z: Números Zuckerman.

A: Números alternos.

Ingrese una opción:

Z

Ingrese el rango: 100 200

En el rango [100,200] hay los siguientes números Zuckerman:

111 112 115 128 132 135 144 175

Menú de opciones:

Z: Números Zuckerman.

A: Números alternos.

Ingrese una opción:

Z

Ingrese el rango: 230 300

En el rango [230,300] hay los siguientes números Zuckerman:

En el rango no existen números Zuckerman

Menú de opciones:

Z: Números Zuckerman.

A: Números alternos.

Ingrese una opción:

A

Ingrese el rango: 60 90

En el rango [60,90] hay los siguientes números alternos:

61 63 65 67 69 70 72 74 76 78 81 83 85 87 89 90

Menú de opciones:

Z: Números Zuckerman.

A: Números alternos.

Ingrese una opción:

A

Ingrese el rango: 150 185

En el rango [150,185] hay los siguientes números alternos:
161 163 165 167 169 181 183 185

Menú de opciones:

Z: Números Zuckerman.

A: Números alternos.

Ingrese una opción:

A

Ingrese el rango: 310 320

En el rango [310,320] hay los siguientes números alternos:

En el rango no existen números alternos

Programa 1: Propuesta de solución - Números Zuckerman y alternos

```
1  #include <stdio.h>
2  #include <math.h>
3
4  /*Prototipos*/
5  void mostrarNumerosZuckerman(int limInf,int limSup);
6  void mostrarNumerosAlternos(int limInf,int limSup);
7  void sacarCifras(int numero,int *unidad,int *decena,int *centena);
8
9  int main(){
10
11     /*Declaración de variables*/
12     int limInf,limSup;
13     char opcion;
14
15     printf("Menú de opciones:\n");
16     printf("Z: Números Zuckerman.\n");
17     printf("A: Números alternos.\n");
18     printf("\nIngrese una opción:\n");
19     scanf("\n%c",&opcion);
20
21     if(opcion=='Z' || opcion=='A'){
22         printf("\nIngrese el rango: ");
23         scanf("%d %d",&limInf,&limSup);
24         if(limInf>0 && limInf<limSup && limSup<1000){
25             if(opcion=='Z')
26                 mostrarNumerosZuckerman(limInf,limSup);
27             else
28                 mostrarNumerosAlternos(limInf,limSup);
29         }
30     } else
31         printf("\nEl rango ingresado es inválido.\n");
32 }
33
34     printf("\nOpción inválida.\n");
35
36     return 0;
37 }
38
39 /*Funciones*/
40 void mostrarNumerosZuckerman(int limInf,int limSup){
41     int i=limInf,unidad,decena,centena,cantDigitos,multiplicacion,cantNumeros=0;
42
43     printf("\nEn el rango [ %d, %d] hay los siguientes números Zuckerman: \n",limInf,limSup);
44     do{
45         sacarCifras(i,&unidad,&decena,&centena);
46         cantDigitos=log10(i)+1;
47         if(cantDigitos==1)
48             multiplicacion=unidad;
```

```

49         else if(cantDigitos==2)
50             multiplicacion=unidad*decena;
51         else
52             multiplicacion=unidad*decena*centena;
53         if(multiplicacion!=0 && i %multiplicacion==0){
54             printf("%d ",i);
55             cantNumeros++;
56         }
57         i++;
58     }while(i<=limSup);
59
60     if(cantNumeros==0)
61         printf("En el rango no existen números Zuckerman\n");
62 }
63
64 void mostrarNumerosAlternos(int limInf,int limSup){
65     int i=limInf,unidad,decena,centena,cantDigitos,validoDosCifras,validoTresCifras,cantNumeros=0;
66
67     printf("\nEn el rango [ %d, %d] hay los siguientes números alternos: \n",limInf,limSup);
68     do{
69         sacarCifras(i,&unidad,&decena,&centena);
70         cantDigitos=log10(i)+1;
71         validoDosCifras=cantDigitos==2 && ((unidad %2==0 && decena %2!=0) || (unidad %2!=0 && decena %2==0));
72         validoTresCifras=cantDigitos==3 && ((unidad %2==0 && decena %2!=0 && centena %2==0) || (unidad %2!=0 &&
73             decena %2==0 && centena %2!=0));
74         if(cantDigitos==1 || validoDosCifras || validoTresCifras){
75             printf("%d ",i);
76             cantNumeros++;
77         }
78         i++;
79     }while(i<=limSup);
80
81     if(cantNumeros==0)
82         printf("En el rango no existen números alternos\n");
83 }
84
85 void sacarCifras(int numero,int *unidad,int *decena,int *centena){
86     (*unidad)=numero %10;
87     numero=numero/10;
88     (*decena)=numero %10;
89     (*centena)=numero/10;
90 }

```

Puede usar cualquier estructura selectiva por lo que el uso de la estructura selectiva múltiple queda a su criterio

No puede usar estructuras iterativas de entrada controlada ni anidadas.