

Introducción a la programación en C

eMe - Escuela Universitaria de Música

Hoja de ejercicios Nro 3

Ejercicio 1

Considere la siguiente secuencia de 13 números,

[0 5 10 2 7 12 4 9 1 6 11 3 8]

La regla de generación de la secuencia (partiendo de 0) puede enunciarse como: sumar 5 y si el resultado es al menos 13, restar 13.

- a) Escriba un programa que genere la secuencia en forma repetida.
- b) Modifique el programa anterior para definir una función que al ser invocada devuelva un número de la secuencia. Al llamar a la función repetidamente se debe obtener la secuencia completa, es decir debe devolver en cada llamada el siguiente número de la secuencia.
- c) Partiendo de un número MIDI, escriba un programa que genere notas MIDI de acuerdo a la secuencia anterior.
- d) Despliegue además de la nota MIDI su correspondiente valor en frecuencia. Implemente una función que al ser llamada reciba el número MIDI y devuelva el valor de frecuencia.

El siguiente es un ejemplo de ejecución, partiendo de la nota MIDI 69,

\$./secuencia

Nota MIDI	Frecuencia(Hz)
69	440.000
74	587.330
79	783.991
71	493.883
76	659.255
81	880.000
73	554.365
78	739.989
70	466.164
75	622.254
80	830.609
72	523.251
77	698.456

Ejercicio 2

La serie de Fibonacci comienza con los números 0 y 1, y cada número subsecuente es la suma los dos números de Fibonacci previos.

[0, 1, 1, 2, 3, 5, 8, 13, 21, ...]

Tiene la propiedad de que la relación de números sucesivos de Fibonacci converge a un valor constante denominado sección áurea (1,618).

a) Escriba un programa que genere en forma iterativa una cierta cantidad de términos de la serie de Fibonacci. Considere como referencia el siguiente ejemplo de ejecución.

```
$ ./fibonacci1
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
```

b) La serie de Fibonacci puede ser definida en forma recursiva de la siguiente forma:

```
fibonacci(0) = 0
fibonacci(1) = 1
fibonacci(n) = fibonacci(n-1) + fibonacci(n-2)
```

Escriba una función que permita implementar la serie de Fibonacci en forma recursiva.

c) Modifique el programa de la parte a) para desplegar el cociente entre números sucesivos de la serie de Fibonacci. Observe que converge a la sección áurea, como en el siguiente ejemplo de ejecución.

```
$ ./fibonacci3
1.000 2.000 1.500 1.667 1.600 1.625 1.615 1.619 1.618 1.618 1.618
```

Ejercicio 3

a) Escriba un programa que genere una secuencia de números enteros consecutivos (partiendo de 1), los multiplique por la sección áurea y tome la parte entera.

b) Genere una secuencia adicional cuyo primer valor sea 1 y luego represente la diferencia entre números sucesivos de la secuencia anterior.

A continuación se presenta un ejemplo de ejecución que despliega las tres secuencias propuestas.

\$./aurea

a)

1	1	1
2	3	2
3	4	1
4	6	2
5	8	2
6	9	1
7	11	2
8	12	1
9	14	2
10	16	2
11	17	1
12	19	2
13	21	2

c) En base a las secuencias anteriores, considere una nueva secuencia que para cada elemento de la secuencia de enteros consecutivos tome el valor 1 si aparece en la segunda secuencia, y 0 en caso contrario.

Sugerencia: puede resultar útil en este caso definir un arreglo con la segunda secuencia y recorrerlo para cada elemento de la secuencia de enteros consecutivos.

d) Por último considere la secuencia que se obtiene restando las dos últimas secuencias generadas y sumando 1.

El siguiente ejemplo de ejecución despliega todas las secuencias propuestas.

\$./aurea4

1	1	1	1	1
2	3	2	0	3
3	4	1	1	1
4	6	2	1	2
5	8	2	0	3
6	9	1	1	1
7	11	2	0	3
8	12	1	1	1
9	14	2	1	2
10	16	2	0	3
11	17	1	1	1
12	19	2	1	2
13	21	2	0	3

Ejercicio 4

a) Escriba un programa que genere una secuencia de notas en forma aleatoria. Para ello implemente una función que defina un sistema de alturas dividiendo cierto intervalo en un número arbitrario de intervalos iguales. La función recibirá como parámetros el intervalo a dividir (ej. 8^a: 2, 5^a de la 8^a: 3), la cantidad de intervalos, la frecuencia a partir de la cual generar las alturas, y un arreglo donde almacenar las frecuencias calculadas. El programa principal sorteará índices dentro del arreglo para elegir las alturas de la secuencia.

A continuación se presenta un ejemplo de ejecución, en el que se generan 5 notas de un conjunto de alturas que divide el intervalo 3 en 13 intervalos iguales, a partir de la frecuencia 440 Hz.

```
$ ./notas1
Frecuencia(Hz)
    730.571
    1024.399
    865.099
    730.571
    865.099
```

Sugerencia: para implementar la función reutilice el código del ejercicio 4 de la hoja de ejercicios Nro 2.

b) Para determinar las duraciones implemente una función que genere proporciones métricas en forma aleatoria y devuelva duraciones en segundos. El numerador de la proporción métrica se obtendrá de la última de las secuencias propuestas en el ejercicio anterior (parte d). El denominador se sorteará entre los números 4, 8, y 16. La función debe recibir como parámetros el valor de un metrónomo en bpm, la cantidad de duraciones a generar y un arreglo donde almacenar las duraciones calculadas.

Considere como referencia el siguiente ejemplo de ejecución.

```
$ ./notas2
Frecuencia(Hz)  Duracion(s)
    1024.399      0.500
    1213.032      1.500
    440.000       0.250
    521.022       1.000
    730.571       3.000
```

Sugerencia: para implementar la función reutilice el código del ejercicio 4 de la hoja de ejercicios Nro 1 y del ejercicio anterior. Puede resultar útil implementar una función adicional encargada de generar la secuencia de numeradores.

Ejercicio 5

Considere el siguiente juego de dados: Un jugador tira dos dados y se suma el valor de cada uno. Si la suma es 7 o 11 en esta primera tirada el jugador gana. Por el contrario si la suma es 2, 3 o 12, el jugador pierde. Si en la primera tirada la suma es 4, 5, 6, 8, 9 o 10, para ganar el jugador debe seguir tirando hasta lograr nuevamente ese valor, que se convierte en los puntos de la jugada. El jugador pierde si su tirada suma 7 antes de alcanzar el valor buscado.

a) Escriba un programa que implemente el juego, de forma de que cada ejecución corresponda a una jugada completa.

Considere como referencia los siguientes ejemplos de ejecución.

```
$ ./dados
Tirada: 1 + 6 = 7
El jugador gana!
```

```
$ ./dados
Tirada: 6 + 6 = 12
El jugador pierde
```

```
$ ./dados
Tirada: 2 + 2 = 4
El valor a alcanzar es: 4
Tirada: 1 + 4 = 5
Tirada: 5 + 5 = 10
Tirada: 3 + 6 = 9
Tirada: 6 + 1 = 7
El jugador pierde
```

```
$ ./dados
Tirada: 3 + 6 = 9
El valor a alcanzar es: 9
Tirada: 2 + 1 = 3
Tirada: 5 + 3 = 8
Tirada: 1 + 4 = 5
Tirada: 5 + 5 = 10
Tirada: 3 + 5 = 8
Tirada: 1 + 1 = 2
Tirada: 2 + 1 = 3
Tirada: 5 + 4 = 9
El jugador gana!
```

Sugerencia: como ejercicio intente plantear el problema usando una estructura de selección `switch` para discriminar los valores de la primera tirada.

b) Modifique el programa anterior para permitir que participen dos jugadores. Se debe ingresar al inicio de la ejecución la cantidad de jugadas que tendrá la partida. Después de la tirada de cada jugador despliegue los datos y pida que se ingrese un valor para continuar (por ejemplo 1).

A continuación se presenta un ejemplo de ejecución para una partida de una sola jugada.

```
$ ./dados2
Ingrese la cantidad de jugadas: 1
```

```
Jugada 1
```

```
Tira jugador A:
```

```
Tirada: 5 + 6 = 11
El jugador gana!
```

```
Ingrese 1 para continuar: 1
```

```
Tira jugador B:
Tirada: 1 + 5 = 6
El valor a alcanzar es: 6
Tirada: 1 + 1 = 2
Tirada: 6 + 5 = 11
Tirada: 4 + 5 = 9
Tirada: 3 + 1 = 4
Tirada: 1 + 6 = 7
El jugador pierde
```

```
Ingrese 1 para continuar: 1
```

Sugerencia: empaquete el código de la parte anterior en una función `int jugada(void)`, que devuelva 1 si el jugador pierde o la cantidad de puntos obtenidos si el jugador gana.

b) Agregue al programa anterior el código necesario para desplegar al final de la partida la cantidad de puntos obtenidos y jugadas ganadas por cada jugador. Represente la cantidad de jugadas ganadas como un barra de asteriscos, como en el siguiente ejemplo de ejecución.

```
$ ./dados2
Ingrese la cantidad de jugadas: 20
```

```
Jugada 1
```

```
Tira jugador A:
Tirada: 4 + 2 = 6
El valor a alcanzar es: 6
Tirada: 6 + 6 = 12
Tirada: 1 + 2 = 3
Tirada: 5 + 1 = 6
El jugador gana!
```

```
.
.
.
```

```
R E S U L T A D O S :
```

```
Puntos:
```

```
Jugador A:      66
Jugador B:      40
```

```
Jugadas ganadas:
```

```
Jugador A:      *****
Jugador B:      *****
```