Tarea de algoritmos, Parte 2

Tulio Muñoz Magaña

September 9, 2020

En los siguientes ejercicios, hacer el planteamiento algebraico y realizar el algoritmo en un diagrama de flujo,además del programa escrito en C.

Planteamientos particulares:

1. De la tarea 1, de diagramas de flujo y algoritmos (donde vienen problemas de físicay probabilidad), traducir los diagramas de flujo a un programa en C, excepto el problema donde hay que usar funciones binomiales o logarítmicasque no hemos visto en clase.

Los códigos se encuentran en la carpeta de códigos y las capturas de pantalla al final de este documento.

2. Se capturan 10 números positivos. La captura termina cuando se introduce un número negativo. Una vez terminada la captura,
a.Calcular el promedio
b.Mostrar el número mayor
c.Mostrar el número menos
Se deben de presentar dos algoritmos, uno con la estructura "while" y otro con la estructura "do while"

Al principio del programa definimos una variable $suma = 0$, a esta variable le vamos sumando los distintos valores que va introduciendo el usuario mientras sean positivos.También al principio del programa definimos una variable max y una min

Entramos en el ciclo. Lo que se hace es designar un variable c para contener los números que vaya introduciendo el usuario. Después de recibir el número, usamos un condicional donde preguntamos si $c \geq 0$, si es así, seguimos el algoritmo, si no, salimos del ciclo.

En la primera iteración del ciclo establecemos $max = c$ y $min = c$. En las siguientes iteraciones cambiamos el valor de max por c si $c > max$, y cambiamos el valor de min por c si $c < min$.

Salimos del ciclo cuando hemos recibido 10 números (salimos antes si el usuario ingresó un número negativo). Al salir del ciclo dividimos *suma* entre i, que es el iterador del ciclo, que corresponderá a las veces que hemos pasado el ciclo. Esto nos dará el promedio de los números ingresados, finalmente imprimimos este promedio, el máximo que está guardado en max y el mínimo que está guardado en min.

3. De que dimensión debe deben ser los lados de un triangulo equilátero si el área es de 20 unidades. Dibujar el triangulode esta dimensión, usando el carácter "*". Si el numero de caracteres utilizados para dibujar el triangulo, es una aproximación al área del triangulo, calcular el porcentaje de error.

Sabiendo que el área del triángulo es 20, utilizamos la fórmula de área de un triángulo equilátero que en este caso es:

$$20 = \frac{\sqrt{3}L^2}{4}$$

Despejando tenemos que:

$$L = \sqrt{\frac{80}{\sqrt{3}}}$$

Este número es aproximadamente 6.7961, que es el lado de éste triángulo. El triángulo equilátero con lados enteros más cercanos a este valor es el de lado 7. Dibujamos entonces el triángulo equilátero con lado de 7 asteriscos, esto lo hacemos con un ciclo que dibuja un renglón por iteración, y dentro de cada iteración utilizamos también ciclos para dibujar los espacios y asteriscos necesarios.

Calculamos el número de asteriscos dibujados, por suma de Gauss, deben ser $\frac{7(8)}{2}$, que es 28. Finalmente calculamos el error e dividiendo el número de asteriscos entre el área dada al principio y lo multiplicamos por 100, es decir:

$$(\frac{28}{20})100 = 140$$

Por lo que el error en este caso es de 140%.

4. Un calculador aritmético simple. El calculador muestra al usuario las operaciones que puede. Estas son
a.Suma
b.Resta
c.Multiplicación
d.División
Cada operación es identificada por una etiqueta numérica o alfabética (como en este caso). Un usuario elige la opción 'b', entonces el calculador le pide los dos números que operará. Después, el calculador muestra el resultado y queda listo

para que el usuario elija una nueva operación. Existe una opción especial que le indica al calculador que se apague.Presentar dos algoritmos, uno con "while"sin usar la estructura de control "switch",y otro con "do while"usando la estructura de control "switch"

Algoritmo switch: $do\{$

Presentamos las opciones al usuario, luego almacenamos lo que nos responde en un caracter c. Luego preguntamos si c es alguno de a, b, c o d, que son los caracteres válidos para operación, si es así, pedimos al usuario que ingrese los números a y b. Evaluamos la función switch, en cada caso, hacemos la operación respectiva e imprimimos el resultado al usuario. En el caso 'S' (que es para salir) no hacemos nada, y en el default decimos que no hemos entendido.

$\}while$(c sea diferente de 'S')

Planteamientos generales:

1. De la tarea 1, de diagramas de flujo y algoritmos (donde vienen problemas de físicay probabilidad), traducir los diagramas de flujo a un programa en C, excepto el problema donde hay que usar funciones binomiales o logarítmicasque no hemos visto en clase.

Los códigos se encuentran en la carpeta de códigos y las capturas de pantalla al final de este documento.

2. Se capturan N números positivos. La captura termina cuando se introduce un número negativo. Una vez terminada la captura,
a.Calcular el promedio
b.Mostrar el número mayor
c.Mostrar el número menos
Se deben de presentar dos algoritmos, uno con la estructura "while" y otro con la estructura "do while"

Pedimos N al usuario.

Definimos una variable $suma = 0$, a esta variable le vamos sumando los distintos valores que va tomando c mientras sean positivos.

Definimos una variable max y una min, en la primera iteración del ciclo establecemos $max = c$ y $min = c$. En las siguientes iteraciones cambiamos el valor de max por c si $c > max$, y cambiamos el valor de min por c si $c < min$.

Al salir del ciclo dividimos $suma$ entre i, que es el iterador del ciclo, que corresponderá a las veces que hemos pasado el ciclo. Esto nos dará el promedio de los números ingresados, finalmente imprimimos este promedio, el máximo

que está guardado en max y el mínimo que está guardado en min.

El ciclo se sigue repitiendo mientras c sea positivo, de lo contrario salimos y calculamos el promedio hasta el número que vayamos.

3. De que dimensión debe deben ser los lados de un triangulo equilátero si el área es de 20 unidades. Dibujar el triangulode esta dimensión, usando el carácter "*". Si el numero de caracteres utilizados para dibujar el triangulo, es una aproximación al área del triangulo, calcular el porcentaje de error.

Sabiendo que el área del triángulo es a, utilizamos la fórmula de área de un triángulo equilátero que es:

$$a = \frac{\sqrt{3}L^2}{4}$$

Despejando tenemos que:

$$L = \sqrt{\frac{4a}{\sqrt{3}}}$$

Utilizamos la función round para encontrar el entero más cercano a este número. Dibujamos entonces el triángulo equilátero con lado de $round(L)$ asteriscos, esto lo hacemos con un ciclo que dibuja un renglón por iteración, y dentro de cada iteración utilizamos también ciclos para dibujar los espacios y asteriscos necesarios.
Calculamos el número de asteriscos dibujados, por suma de Gauss, deben ser $\frac{round(L)(round(L)+1)}{2}$.
Finalmente calculamos el error e dividiendo el número de asteriscos entre el área dada al principio y lo multiplicamos por 100, es decir:

$$(\frac{round(L)(round(L)+1)}{2a})100 = e$$

Finalmente imprimimos e.

4. Un calculador aritmético simple. El calculador muestra al usuario las operaciones que puede. Estas son
a.Suma
b.Resta
c.Multiplicación
d.División
Cada operación es identificada por una etiqueta numérica o alfabética (como en este caso). Un usuario elige la opción 'b', entonces el calculador le pide los dos números que operará. Después, el calculador muestra el resultado y queda listo para que el usuario elija una nueva operación. Existe una opción especial que le indica al calculador que se apague.Presentar dos algoritmos, uno con "while"sin usar la estructura de control "switch",y otro con "do while"usando la estructura de control "switch"

Algoritmo sin switch:

En este algoritmo usamos una variable de control v que será 1 si c es alguno de a, b, c o d, y 0 si no. Esto es para saber si el usuario ingresó un caracter válido paraq realizar una operación.

Al principio del algoritmo, hacemos $v = 0$. Presentamos las opciones al usuario, luego almacenamos lo que nos responde en un caracter c. Luego preguntamos si c es alguno de a, b, c o d, que son los caracteres válidos para operación, si es así, establecemos $v = 1$ y entramos en el ciclo.

while(c es diferente de 'S'){

Si $v = 1$ pedimos al usuario que ingrese los números a y b. Luego entramos a una cadena de 5 condicionales anidados, donde se verifica se c es $'a'$, luego si c es $'b$, etc. Dentro de cada secuencia hacemos la operación correspondiente, o bien imprimimos que no hemos entendido (si $c =' S'$ no hacemos nada). Después de esto volvemos a preguntar si c es 1, si es así imprimimos el resultado de la operación. Antes de cerrar el ciclo, volvemos a hacer $v = 0$ y volvemos a pedirle al usuario que nos dé c. Una vez hecho esto volvemos a cambiar v a 1 si c es un caracter válido para operación.

}

Termina el algoritmo.