

Trabajo para Abril-Mayo 2020.

2º Bachillerato

Programación en C. TIC II

Siguiendo con las tareas que tenemos pendientes en nuestra asignatura, vamos a acabar el lenguaje de programación C con la realización de los últimos programas. Mi intención es que sea la última tarea que os envíe, dada la situación en la que nos encontramos (repito, mi intención....)

A diferencia del anterior trabajo, os detallaré cada programa para que no haya dudas (o eso espero). Cada ayuda que me pidáis (de C) se descontará en la puntuación de cada programa aludido si considero que estaba ya lo suficientemente explicado.

Como véis, empiezo donde acabamos, con el programa número diez.

Ánimo y paciencia

10.- Programa que calcule las variaciones de 2 números m y n. La supuesta fórmula de las variaciones es: $Variaciones = m! / (m-n)!$

Ambos números (m y n) deben ser positivos y tenéis que controlarlo (con do/while).

m debe ser mayor que n y también tenéis que controlarlo.

Pistas: no hay for anidados. Primero calcular el numerador (el factorial de un número ya lo hemos hecho en el programa número 9) y después el denominador.

Así, tenemos 2 FOR.

Ejemplo:

Dime m 4

Dime n (m debe ser mayor que n) 2

Las variaciones de 4 elementos tomados de 2 en 2 es 12

11.- Programa que calcule las combinaciones de 2 números m y n. La supuesta fórmula de las combinaciones es: $\text{Combinaciones} = m! / (n! * (m-n)!)$

Ambos números (m y n) deben ser positivos y tenéis que controlarlo (con do/while).

m debe ser mayor que n y también tenéis que controlarlo.

Pistas: no hay FOR anidados. Primero calcular el numerador (el factorial de un número ya lo hemos hecho en el programa número 9) y después, con el denominador, debéis dividirlo en dos partes, de modo que debemos calcular por separado los dos factoriales del cual se compone. Así, tendremos tres FOR.

Ejemplo:

Dime m 4

Dime n (m debe ser mayor que n) 2

Las combinaciones de 4 elementos tomados de 2 en 2 es 6.00

12.- Programa que calcule las permutaciones de 1 números dado que tendremos que recoger. La supuesta fórmula de las permutaciones es: $\text{Permutaciones} = n! / (n! * (n-1)! * (n-2)! * \dots * 2! * 1!)$

El número (n) debe ser positivos y tenéis que controlarlo (con do/while)

Pistas: Primero calcular el numerador (el factorial de un número ya lo hemos hecho en el programa número 9) y después para calcular el denominador deberemos usar un FOR anidado. El FOR exterior indicará el número (del cuál queremos hacer el factorial) que será “enviado” al FOR interior para que éste realice el factorial de dicho número. Dentro de éste último FOR sólo habrá una sentencia de asignación que vaya multiplicando todos los factores.

Ejemplo:

Dime n (debe ser mayor que 0) 4

Las permutaciones de 4 elementos tomados de 4 en 4 es 0.083333

13.- Dejamos el maravilloso mundo de las matemática para realizar un programa corto: Debéis realizar un programa que escriba números desde 1 hasta un número que indicará el usuario. Si ese número es mayor que 200, deberá escribir todos los números del 1 al 200 y después romper el bucle (justo después de escribir 200) y mostrar (**insisto: Después**): "No se permiten valores por encima de 200". Si se da el caso que es menor que 200, simplemente se mostrará del 1 hasta ese número.

Pistas: No se debe controlar el número, el usuario lo escribe y ya está. Sólo habrá un bucle FOR que irá desde 1 hasta el número que se recoja. Dentro de él se irán mostrando todos los números. Si se da la circunstancia que se llega al 200, habrá que controlar esta situación con una condición dentro del FOR (no en la sentencia del FOR....sino dentro). Pista plus: habrá que usar la sentencia "break"

Ejemplo A:

```
Dime un numero (menor que 200) 56
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
```

Ejemplo B:

14.- Mostrad una semi-pirámide así

```
1
12
123
1234
12345
123456
1234567
12345678
123456789
```

Pistas: Habrá únicamente dos variables, que serán los contadores para los dos FOR que se usarán en el programa, por supuesto, anidados. El FOR exterior significará el número de línea que le manda al FOR interior (en nuestro caso son 9). Éste imprimirá los números del bucle (que serán los números que queremos mostrar en cada línea), que irá desde 1 hasta el número que precisamente “le pasa” el primer FOR. Cuidado con el salto de línea.

15.- El programa mostrará la secuencia 0, 1, 3, 6, 10, 15, 21..... hasta el número (límite) que quiera el usuario (siempre positivo, por lo que debéis controlar eso con do/while).

Pistas: se necesitarán 3 variable. Una para recoger el límite, otra como contador del bucle FOR (aquí no hay más que un único bucle) y otra para ir avanzando en la secuencia. Daos cuenta que se le va sumando a cada número el 1, el 2, el 3, etc..

Así:

0+1=1 **1+2=3** **3+3=6** **6+4=10** **10+5=15** **15+6=21**

Como véis, los números en negrita serán el contador del bucle.

Ejemplo:

16.- Haced un programa que pide un número y diga si es primo o no

Deberá realizarse con FOR (sólo habrá uno) y obligatoriamente se usará break para romperlo cuando se dé la circunstancia que no es primo (un número es primo cuando sólo tiene dos divisores, él mismo y el uno). Por tanto si lo vamos dividiendo entre 2, 3, 4, 5, 6, etc... y ya con el 2 vemos que el resto de la división es cero, realizamos un break.

Ejemplo A:

Dame un número 437

El número 437 NO es primo

Ejemplo A:

Dame un número 431

El número 431 es primo

17.- Programa que pida un número y muestre todos los primos entre el 1 y ese número recogido.

Deberá realizarse con dos FOR (evidentemente, anidados) y obligatoriamente se usará break para romper el FOR interior cuando se dé la circunstancia que el número que “pasa” el FOR exterior al interior no es primo.

Así, el FOR exterior será un bucle que irá desde el 1 hasta el número que ha introducido el usuario. El FOR interior será un bucle similar con el que habremos solucionado el programa anterior y contendrá un break.

Ejemplo:

```
Dame un numero 434

Los número primos de 1 hasta 434 son 1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 16
3 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 27
1 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 39
7 401 409 419 421 431 433

-----
(program exited with code: 0)
Press return to continue
```