

Examen parcial núm. 2

13 nov 2013

PRIMERA PARTE (ESCRITA)

1. Contesta las siguientes preguntas en relación con un arreglo denominado `table`.
 - a) Declara `table` como un arreglo de números enteros que representa una tabla con 5 renglones y 5 columnas. Declara una constante simbólica `SIZE` con el valor 5 y usala en la declaración de `table`.
 - b) ¿Cuántos elementos contiene el arreglo `table`?
 - c) Utiliza una estructura de repetición `for` para asignar a cada elemento de `table` la suma de sus subíndices. Por ejemplo, el valor de `table[2][3]` debe ser 5. Supon que han sido declaradas las variables enteras `i`, `j` y usalas como variables de control.
 - d) Utiliza una estructura de repetición de tu preferencia (`for`, `while`, `do-while`) para imprimir los valores del arreglo `table`.

2. Encuentra el error en cada uno de los siguientes segmentos de programa y corrige el error.

a)

```
#define SIZE 10;
```

b)

```
SIZE = 10
```

c)

```
int b[10] = {0}, i;  
for(i=0; i <=10; i++)  
    b[i] = 1;
```

d)

```
int a[2][2] = {{1,2},{3,4}};  
a[1,1] = 56;
```

3. Realiza una prueba de escritorio del siguiente programa:

```
#include<stdio.h>
int recursivo(int, int);

int main()
{
    int x,y;
    printf("Ingresa dos enteros:");
    scanf("%d%d", &x, &y);
    printf("El resultado es %d", recursivo(x,y));
    return 0;
}

/*El valor de b debe de ser positivo*/
int recursivo(int a, int b)
{
    if(b == 1)
        return a;
    else
        return a + recursivo(a, b-1);
}
```

SEGUNDA PARTE (PRÁCTICA)

1. Escribe un programa que imprima el valor más pequeño de las entradas de un arreglo de enteros declarado como `int table[5][5]`. El programa tendrá:
 - Una función `int general()`; la cual regresa un número entero aleatorio en el rango -100 y 100.
 - En la función principal utiliza la función anterior para iniciar el arreglo `table`.
 - Una función `void imprime(int[][5])`; la cual imprima los elementos de un arreglo de 5×5 .
 - Una función `int menor(int[][5])`; la cual regresa el elemento de menor valor de un arreglo.

El programa debe imprimir el arreglo generado aleatoriamente y luego imprimir el elemento de menor valor del mismo.

2. Se dice que un entero $n > 1$ es *perfecto* si es igual a la suma de sus divisores positivos, incluyendo a 1 pero excluyendo a n mismo. Por ejemplo, 6 es el primer número perfecto ya que $6 = 1 + 2 + 3$ y 2, 3, 4, 5 no son perfectos ($2 \neq 1$, $3 \neq 1$, $4 \neq 1 + 2$, $5 \neq 1$). Escribe una función `perfecto(n)` que determina si n es un número perfecto. Utiliza esta función en un programa que imprime todos los números perfectos entre 1 y 1500.
3. Escribe un programa que inserta una cadena dentro de otra en un punto especificado (la numeración empieza en 0).

Por ejemplo:

Entrada	Salida
holaestas como 4	holacomoestas
estas como 0	comoestas

Corre tu programa con al menos tres entradas diferentes y completa la siguiente tabla

Entrada	Salida

Extra crédito: inserta espacios en blanco antes y después de la segunda cadena.

Por ejemplo:

Entrada	Salida
holaestas como 4	hola como estas
estas como 0	como estas