

Tecnólogo en Informática - Principios de Programación - curso 2007

Práctico 9: arreglos

Ejercicio 1

Para las siguientes declaraciones, decir el tamaño del arreglo y el rango en el cual varían los índices.

```
i.    int a[10];
ii.   char arr_car[8];
iii.  #define MAX 250
      .....
      char arreglo[MAX];
```

Ejercicio 2

Para el programa que se da a continuación, decir cuáles asignaciones son inválidas (no compilan) o inapropiadas. Justificar.

```
#define MAX 20
main()
{
    char arreglo[MAX];
    char c1, c2='0';
    int n;

    arreglo[0] = n;
    arreglo[1] = c2;
    arreglo[MAX] = '$';
    for (n=0; n <= MAX-1; n++)
        arreglo[n] = c1;
    for (n=1; n<= MAX; n++)
        arreglo[n] = c2;
    for (n=1; n<= MAX; n++)
        arreglo[n-1] = '?';
    for (n=1; n<= MAX-1; n++)
        arreglo[n] = c2;
}
```

Ejercicio 3

Escribir un programa para leer una secuencia de 10 números de la entrada y desplegarlos en orden inverso al orden en el cual fueron leídos. El programa debe ser fácilmente adaptable a otras cantidades de números.

Ejercicio 4

Escribir un programa para leer una secuencia de números positivos y desplegarlos en orden inverso al orden en el cual fueron leídos. La cantidad de números ingresados es $\leq N$, siendo N una constante entera. El fin de la secuencia está indicado por el valor **-1** (centinela).

Ejercicio 5

Se declaran los siguientes arreglos para representar cadenas de caracteres de largo **M** y **N**, con **M < N**:

```
char arrM[M];  
char arrN[N];
```

Escribir un programa que lea dos cadenas de la entrada, de largo **M** y **N** respectivamente, y determine si la cadena de largo **M** está incluida dentro de la cadena de largo **N** (es subcadena). El programa debe funcionar para cualquier valor positivo que puedan tomar las constantes **M** y **N**, siempre que se cumpla **M < N**.

Ejercicio 6

Se declaran los siguientes arreglos para representar cadenas de caracteres de largo **M** y **N**:

```
char arrM[M];  
char arrN[N];
```

Escribir un programa que lea dos cadenas de la entrada, de largo **M** y **N** respectivamente, y determine si algún carácter de la primera cadena ocurre en la segunda cadena.

Ejercicio 7

Se declaran los siguientes arreglos para representar cadenas de caracteres de largo **M** y **N**:

```
char arrM[M];  
char arrN[N];
```

Escribir un programa que lea dos cadenas de la entrada, de largo **M** y **N** respectivamente, y determine si todos los caracteres de la primera cadena ocurren en la segunda cadena (en cualquier posición y orden).

Ejercicio 8

Escribir un programa que lea de la entrada estándar una secuencia de enteros positivos que termina con el centinela **-1** y un valor real. El programa debe imprimir el resultado de evaluar $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$, en donde los coeficientes del polinomio son los enteros leídos de la entrada y **x** el valor real.

¿Existe alguna ventaja en aplicar la regla de Horner para la evaluación del polinomio?

regla de Horner: $p(x) = (\dots((a_n x + a_{n-1}) x + a_1) x + \dots a_1) x + a_0$

Ejercicio 9

Escribir un programa que:

- lee de la entrada un entero de **N** dígitos que contiene un número representado en una base entre **2** y **10**,
- lee la base correspondiente al número leído y
- despliega el equivalente en base **10**.

N es una constante que puede ser fijada en cualquier valor entero menor que **10**.

Asumir que las entradas son válidas, es decir, si se ingresa un número en base **b**, los dígitos utilizados estarán entre **0** y **b-1**.

Recordar que la fórmula para calcular la representación decimal de un número es:

$$d1 * base^{N-1} + d2 * base^{N-2} + \dots + dN$$

(*d1*, ..., *dN* son los **N** dígitos del número leído.)

Ejemplo para N=5

```
Numero: 32021  
Base: 4  
Resultado: 905
```