

## Práctico 9 Soluciones

### 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.

```
#define M 10
```

```
#define N 7
```

```
void ej6(){
    //ejercicio 6 práctico 9
    char arrN[N];
    char arrM[M];
    int i=0;
    int j=0;
    bool encuentre=false;
    printf("Ingrese la primera cadena: \n");
    scanf("%s",arrN);
    printf("Ingrese la segunda cadena: \n");
    scanf("\n%s",arrM);
    while(i<strlen(arrM) && !encontre){
        j=0;
        while(!encontre && j<strlen(arrN)){
            if(arrM[i]==arrN[j]){
                encuentre=true;
            }
            j++;
        }
        i++;
    }
    if(encontre){
        printf("Si hay una ocurrencia por ejemplo la letra %c \n",arrM[i-1]);
    }
}

int potencia(int base,int exponente){
    int res=1;
    for(int i=0;i<exponente;i++){
        res=res*base;
    }

    return res;
}
```

### 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?

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

```
#define MAX 100
```

```
void ej8(){

    int coefs[MAX];
    //scanfEnteros(coefs,MAX);
    //ejercicio 8 práctico 9
    //Regla de horner
    //Ingresamos los coeficientes de uno en uno apreteando enter, en orden creciente de
grado
    int x=0;
    int resultado=0;
    int i=0;
    int entrada=0;
    printf("Ingrese un valor para x:");
    scanf("%d",&x);
    printf("Ingrese los coeficientes separados por enter e ingrese -1 para finalizar:\n");
    printf("(Los coeficientes se ingresan en orden creciente)\n");
    while(entrada!=-1){
        printf("Ingrese coeficiente\n");
        scanf("%d",&entrada);
        coefs[i++]=entrada;

    }
    //regla de horner
    for(int h=i-2;h>=0;--h){
        resultado=resultado*x + coefs[h];
    }

    printf("\nEl resultado de evaluar el polinomio con x = %d es %d\n",x,resultado);

}
```

### 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:

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

( $d1, \dots, dN$  son los  $N$  dígitos del número leído.)

Ejemplo para  $N=5$

**Numero:** 32021

**Base:** 4

**Resultado:** 905

```
void ej9(){

    //ejercicio 9 práctico 9
    //convierte un numero en una base menor a 10 a nuemro decimal(base 10)
    int nbase10=0;
    unsigned short base;
    int largo;
    char numero[N];
    printf("Ingrese el numero: \n");
    scanf("%s",numero);
    printf("Ingrese la base: \n");
    scanf("%d",&base);

    largo=strlen(numero)-1;
    for(int i=largo;i>=0;i--){
        nbase10+=((numero[i]-48)*potencia(base,largo-i));//resto 48 ya que 48 es el
        codigo ascii del digito 0

    }
    printf("El numero %s en base 10 es %d\n",numero,nbase10);

}
```