CENTRO SUPERIOR DE INFORMÁTICA
Departamento de Estadística, I. O. y Computación

Teoría de Autómatas y Lenguajes Formales

Problemas de Programación en C

1.– Punteros. Explicar el significado de cada una de las declaraciones siguientes:
a) int *p;
b) int *p[10];
c) int (*p)[10];
d) int *p(void);
e) int p(char *a);
f) int *p(char *a);
g) int (*p)(char *a);
h) int (*p(char *a))[10];
i) int p(char (*a)[]);
j) int p(char *a[]);
k) int *p(char a[]);
l) int *p(char (*a)[]);
m) int *p(char]a[]);
n) int (*p)(char (*a)[]);
o) int *(*p)(char (*a)[]);
p) int *(*p)(char *a[]);
q) int (*p[10])(void);
r) int (*p[10])(char a);
s) int *(*p[10])(char a);
t) int *(*p[10])(char *a);

2.– Implementar una función a la que se le pasan tres punteros a matriz y que devuelva en la tercera matriz el producto de las dos primeras. Implementar la misma función pero usando sólo punteros.

3.– Construir una función que inserte elementos en una lista doblemente encadenada y otra que borre un nodo de una lista doblemente encadenada.

4.– Implementar en C una pila utilizando memoria dinámica.

5.– Construir una función a la que se le pase como entrada una expresión infija y devuelva como resultado la misma expresión escrita en notación postfija.

6.– Diseñar una función que inserte elementos en un árbol ordenado y otra que borre un determinado valor de un árbol ordenado.

7.– Funciones de manejo de cadenas. Diseñar las siguientes funciones.

void Scat(char *str1, char *str2) Concatenará la string 2 al final de la string 1.

char *Sleft(char str, int n) Devuelve una string que consta de los ’n’ primeros caracteres de la string ’str’.

char *Sright(char str, int n) Devuelve una string que consta de los ’n’ últimos caracteres de la string ’str’.

Para implementar las funciones anteriores no se puede usar las funciones de la librería "string.h". Tener en cuenta que las strings en C acaban con un caracter ’\0’ al final.
Implementar las rutinas anteriores de forma que las cadenas se guarden en memoria del mismo modo que se hace en Turbo Pascal, es decir, usando el primer byte para indicar el número de caracteres de la string.

8.– Ficheros. Usando las rutinas de alto nivel, leer un fichero de texto y generar una lista encadenada con la frecuencia de aparición de las palabras en el texto. Usando las rutinas de bajo nivel implementar una función que cuente el número de líneas y el número de apariciones de cada uno de los caracteres en un fichero de texto.

9.– Implementar una función que dada una matriz de adyacencia con costes de un grafo calcule el camino mínimo entre dos vértices.

10.– Diseñar en C un programa que encripte un fichero de texto. El método de encriptado consistirá en sustituír cada caracter original por el resultado de hacer un XOR del código ASCII de ese caracter con el caracter correspondiente de la clave secreta. Los caracteres de la clave secreta se recorrerán cíclicamente: 1, 2, ..., N, 1, 2, ...
El mismo programa, realizando la misma operación deberá permitir decodificar un fichero codificado, dada la clave que se ha utilizado.

11.– Diseñar un programa que sustituya en un fichero de texto todas las palabras por las mismas palabras con las letras escritas al revés.

12.– Buscar una tabla de conversión de letras del alfabeto a alfabeto morse y realizar un programa que escriba en morse un fichero de texto que tome como entrada.

13.– Un programa contiene las siguientes declaraciones:

```
char u, v = 'A';
char *pu, *pv = &v;
...
*pv = v + 1;
u = *pv + 1;
pu = &u;
```

Supóngase que cada carácter ocupa 1 byte de memoria. Si el valor asignado a u se almacena en la dirección F8Ch y el valor asignado a v se almacena en F8Dh, entonces:
a) ¿Qué valor es representado por &v?
b) ¿Qué valor es asignado a pv?
c) ¿Qué valor es representado por *pv?
d) ¿Qué valor es asignado a u?
e) ¿Qué valor es representado por &u?
f) ¿Qué valor es asignado a pu?
g) ¿Qué valor es representado por *pu?