Examen convocatoria de Junio de 2000

Nombre:_____

1. (2.5 puntos) Dí si son verdaderas o falsas (V o F) cada una de las afirmaciones siguientes acerca de los conceptos que se señalan con un punto (Nota: los fallos descuentan puntos)

☐ Examinando el MAR (Registro de Direcciones de la CPU), vemos que tiene 20 bits. Esto nos indica que la RAM tendrá exactamente una extensión de 2^{20} bytes (ni más ni menos).

☐ Los modelos de computación describen procesos mecánicos para tratar la información

☐ Un programa tiene dos partes bien diferenciadas, el algoritmo y el proceso mecánico que se ejecuta.

☐ El uso adecuado del diseño descendente o 'Top-Down' en programación estructurada, genera programas que son directamente modularizables.

☐ Un programa que no tenga sentencias iterativas ejecutará cada una de sus sentencias como máximo una vez.

☐ Unicamente la función `main ()` en C tiene la autorización para acceder a variables locales de otras funciones en el programa

☐ Si en un compilador el tipo de dato `int` usa 16 bits, se producirá un desbordamiento de una variable de este tipo si intento asignarle un valor entero mayor que $(2^{16}) / 2$

☐ Los parámetros formales de una función deben de pasarse por valor mientras que los parámetros reales se deben pasar por referencia

☐ El encapsulamiento de los datos en un subprograma significa que las variables locales de dicho subprograma tienen como ámbito el propio subprograma.

☐ El error en ejecución conocido como "desbordamiento de Pila" se produce en la fase de plegado de funciones recursivas mal diseñadas.

☐ El tipo de una variable indica exclusivamente el espacio en memoria que ocupa esa variable

☐ Al decir que C favorece la programación modular, nos referimos a que el programa en memoria tiene 4 módulos (Segmento de código, Seg. de datos, Pila y Montículo)

☐ Si tenemos que pasar como argumento a un subprograma una estructura de datos de gran tamaño, es mejor hacerlo por valor, pues sólo pasamos los valores y no por referencia donde copiamos, además de los valores, información adicional.

☐ El acceso a un elemento de una matriz es secuencial. Por ello, para acceder a un elemento suele utilizarse un doble bucle `for` anidado

☐ No es posible acceder de manera directa a los datos de un fichero cuyo soporte es una cinta magnética y el lector es una única cabeza que discurre sobre la cinta.

2. (1.5 puntos) Relaciona los conceptos de la columna izquierda con los de la derecha **y explica el porqué** de la relación

A- Diseño por refinamiento sucesivo (Top-Down)	1. Interface
B- Uso de un conjunto pequeño de sentencias de control	2. Encapsulamiento de datos
C- Concepto de ámbito de las variables	3. Modularización de los programas
D- Recursividad	4. Uso de la pila del programa
E- Uso de parámetros en los subprogramas	5. Programas comprensibles

Empieza cada contestación de esta forma: Relaciono ?? con ?? porque…

3. (2.5 puntos) Escribe una función en C que compruebe si una cadena de caracteres está contenida en otra cadena de caracteres. Por ejemplo, la cadena

m	a	r	\0	

Esta contenida en la cadena

A	m	a	r	i	l	l	o	\0	

La función tendrá como argumentos la cadena que tengo que buscar (llámala "patron") y la cadena donde busco a la anterior llámala "fuente". Debe distinguir (para mayor facilidad vuestra) entre mayúsculas y minúsculas (es decir "mar" es distinto de "Mar")
y debe devolver un valor entero que será
1 si la cadena patrón se encuentra dentro de la cadena fuente
0 si no es así

4. (2 puntos) Un agente secreto tiene un medio de pasar mensajes cifrados que es el siguiente. El receptor del mensaje posee un enorme fichero texto "`sopa.txt`" que contiene letras sin ningún orden aparente. El agente pasa al receptor un vector de, **como máximo, 200 enteros largos** (`long`) en este vector estan almacenadas las posiciones donde se encuentran en `sopa.txt` las letras que forman el mensaje. El primer número del vector es la posición donde se encuentra la primera letra **a partir del comienzo del fichero**. El resto (que pueden ser números negativos o positivos) muestran la posición de la siguiente letra **desde la posición donde actualmente se encuentra el cursor**, hasta que se encuentra un **0, que indica el final de la serie de números** que codifica el mensaje. Implementa una función que pasándole como argumento un vector de 200 long, abra el fichero "`sopa.txt`" e imprima por pantalla, letra a letra, el mensaje.

5. (1.5 puntos). Indica los errores **semánticos y de uso** (no sintácticos) de estos códigos en C

```
//programa que invierte la secuencia de
//letras introducidas por teclado hasta
//que se pulsa el carácter '@'
void palindroma(void){
  char a;
 a= getch();
 printf("%c\n", a);
 if a!='@'
    palindroma();
}
```

```
void leer(char *nombrefich){
FILE * fp;
 char a;
 do{
    fp= fopen(nombrefich, "r");
    fread(&a, sizeof(char),5,fp); //leo 5 caracteres
   printf("%c\n", a);
    }
 while(!feof(fp);
 fclose();
}
```

```
char linea[30]; // suponer que el vector esta
                //correctamente inicializado
char a;
int i=0;
for(i=0;i<30;i++)
   printf("%c",linea[i];
i=0;
while((linea[i]!='z') || (linea[i] !='\0')){
 printf("%c", linea[i]);
 i++;
}
```

Examen convocatoria de Junio de 2000

Opcional:

1) (0.3 puntos)

El tipo de dato `short` ocupa en una arquitectura concreta 1 byte

Indica el valor en base 10 de las siguientes representaciones en base 2. Nota: los valores negativos estan en complemento a 2

a) 00110110

b) 10000001

2) (0.3 puntos)Pasa a base 10 los siguientes números en diferentes bases:

a) 055 (octal)

b) 0x1A97 (hexadecimal)

Pasa a binario los siguientes números:

a) $3456)_{10}$

b) 0xFA67 (hexadecimal)

3)(0.4 puntos) Tenemos una matriz `M` de dimensión nXn de enteros ya inicializada y una variable `int num` tambien inicializada. Los siguientes algoritmos buscan a num en M y devuelven 1 si lo encuentran y 0 si no está

```
 int i,j;
int encontrado=0;
for(i=0;i<n;i++)
  for(j=0;j<n;j++)
    if M[i][j] == num
       encontrado=1;
return(encontrado)
```

```
int i=0,j=0;
int encontrado=0;
do{
  do{
    if M[i][j] ==num
      encontrado=1;
    j++
   }
  while((j<n) &&(encontrado ==0);
  i++
 }
while((i<n) &&(encontrado ==0)
```

a) Si nos situamos en el peor caso (num no está en M) ¿Cuál sería el coste de cada algoritmo?

b) Si nos encontramos en el mejor caso (num esta en M[0][0])) ¿Cuál sería el coste de cada algoritmo?