
Examen convocatoria de Febrero 2000

Nombre: _____

1. (3 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)

• Según el concepto de información definido por Shannon:

La elección del alfabeto en el que está codificado el mensaje influye en la cantidad de información que transmitimos por símbolo.

La información de un mensaje es medida por la longitud del mínimo algoritmo que reproduce ese mensaje

El mensaje “Se quema nuestra casa” tiene mayor información que el mensaje “Ayer llovió” ya que el primero es más importante que el segundo

• Los computadores basados en el Modelo de Von Newmann

pueden direccionar tantas celdas de memoria como número de canales (1 canal = 1 bit) tenga el bus de direcciones.

son la implementación física y fiel de la Máquina de Turing

juntan las instrucciones entendibles por la CPU y los datos que manejan estas intrucciones en la memoria RAM, constituyendo un programa

Usan un conjunto muy grande de instrucciones entendibles por la CPU porque están diseñadas para abordar problemas de muchos tipos diferentes

• Un algoritmo

No tiene por qué ser un proceso mecánico ya que por ejemplo Euclides no empleó ninguna máquina para definir su célebre algoritmo del máximo común divisor.

No siempre es computable

Es un proceso finito en extensión y en tiempo de ejecución, bien definido y efectivo

• Un tipo de dato

indica única y exclusivamente el espacio de memoria que ocupa una variable

define claramente el ámbito de uso de la variable en el programa

puede ser redefinido en su totalidad por el usuario gracias a los constructores de tipo

puede ser cambiado a otro tipo distinto gracias a la coerción de tipos (cast) que ofrecen lenguajes como C

• La programación estructurada

Genera programas modulares gracias al uso de la metodología de diseño descendente (Top-Down)

Genera programas donde el control del flujo de la información está regulado por unos pocos tipos de sentencias de control.

- Permite la ejecución secuencial de sentencias pertenecientes a distintos módulos mediante el uso de saltos incondicionales de un módulo a otro.
 - El paso de parámetros por referencia a un subprograma
 - Es muy eficiente en C y en otros lenguajes (es decir consume poco tiempo) ya que únicamente hay que copiar la referencia (dirección) de la variable que pasamos.
 - Es muy eficiente puesto que, al no tener que copiar la variable, no utiliza la pila del programa
 - Es posible aplicarlo a las estructuras (struct) del lenguaje C ya que su constructor de tipos permite la operación “obtención de dirección” (&)
 - Las variables declaradas en C y otros lenguajes imperativos dentro de un subprograma
 - Son locales al código del subprograma (es decir sólo pueden ser usadas por el código de la función) debido a que se almacenan en la pila del programa.
 - Con los ficheros
 - Es posible trabajar con ellos a nivel lógico y a nivel físico en programas codificados en lenguajes como C, prefiriendo el nivel lógico por ser más cómodo.
2. (2 puntos) Encuentra los errores semánticos (de concepto, de uso, de ver si hace algo útil) en los siguientes códigos en C. Señálalos con un número y en la zona de respuestas pon el número y la explicación.

Puede haber más de un error en alguno.

```
#define FACTOR 50
int *calculo(int *N){
int aux=0;
int i;
for(i=0;i<FACTOR;i++){
*N = (*N) + i;
aux = aux+ (*N);
}
return(&aux);
}
void main(){
int *P, n=30;
p=calculo(&n);
printf("%d", *p);
}
```

```
#define TOPE 10
void recursiva(int N){
if(N<=10)
recursiva(N+1);
else
recursiva(N-1);
}
void main(){
int dato;
printf("Inicio Secuencia");
recursiva(dato);
printf("\nAcaba secuencia");
}
```

```
struct dato{
int campol;
float campo2;
};
void main(){
FILE *fp;
fp=fopen("datos.dat", "rb");
do{
fread(&dato, sizeof(struct dato),
1, fp);
printf("%d", dato.campol);
}
while(!feof(fp));
fclose(fp);
}
```

```
#define TAM 50
void copia(char *p, char *q){
p= p+(TAM-1);
do{
*p = *q;
p--;
q++;
}
while(*q!='\0');
*p = *q;
}
void main(){
char cadenal[TAM];
char cadena2[TAM];
inicializa(cadena2); /*supon
que esta función esta definida
y que inicializa la cadena de
caracteres*/
copia(cadenal, cadena2);
printf("%s", cadenal);
}
```

Nombre: _____

3. Un mapa de la ciudad de Valencia esta dividido en sectores formando una cuadrícula de 10 x 10 sectores. Las filas están ordenadas con las letras de la A a la J y las columnas con los números del 0 al 9. En cada sector se tiene la siguiente información: nombre del sector, una variable entera que vale 1 ó 0 según el sector forme parte de un camino por la ciudad o no; otra variable entera que indique si ese sector ha sido ya recorrido o no, una variable real que indica la longitud máxima del sector y otra variable entera que indica con un 1 o un 0 si hay una gasolinera o no.
- (0.5 puntos) declara en C una estructura de datos que contenga toda esta información.
 - (1 punto) Imagina que hemos inicializado esa estructura de datos con un camino y poniendo la información visitado a 0 en todos los sectores. Escribe una función que tomando como argumento las coordenadas del sector de comienzo del camino, lo recorra sacando por pantalla las coordenadas de los sectores por los que atraviesa el camino. Nota: el camino es lineal, es decir, no hay bifurcaciones.
 - (1 punto) Modifica la función anterior para que devuelva al final del recorrido cuál es la distancia máxima posible en ese recorrido sin encontrar gasolineras.
4. Sea “enteros.dat” un fichero que contiene números enteros positivos.

```
0 #define PASO_ATRÁS -2
1 void procedimiento1(){
2 FILE *fp, *fk;
3 long i, N;
4 int dato;
5 if( ((fp=fopen("enteros.dat", "rb"))==NULL) ||
6     ((fk=fopen("nuevo.dat", "wb"))==NULL) ){
7     printf("Error abriendo ficheros");
8     exit(1);
9 }
10 fseek(fp, 0, SEEK_END);
11 N=ftell(fp); /*devuelve el número de bytes
                que hay hasta la posición
                actual del localizador de
                posición del fichero*/
12 N = N / sizeof(int);
13 for(i=0; i<N; i++){
14     fread(&dato, sizeof(int), 1, fp);
15     fwrite(&dato, sizeof(int), 1, fk);
16     fseek(fp, PASO_ATRAS, SEEK_CUR);
17 }
18 fclose(fp);
19 fclose(fk);
20 }
```

- (1 punto) Explica en detalle qué hace este procedimiento en C. Centra tu explicación en los pasos 12,14,15 y 16
- (1.5 puntos) Modifica el código para que siga haciendo lo que hace y, además, si el dato leído es un número par, se copie tantas veces en el nuevo fichero como divisible sea por 2 (por ejemplo si el dato es 8 se debería copiar tres veces). Para ello, escribe en el recuadro blanco, el número de línea donde empezarías a insertar tu nuevo código, suponiendo que la instrucción que ocupa esa línea pasa a la anterior.

nº línea →
código:

5. a) (0.5 puntos extra) Da los valores en código decimal de los siguientes números
0x A19D 2) 10111001 3) 54.47_8 4) 10111.0101
- b) (0.5 puntos extra) Ordena de menor a mayor los órdenes de coste siguientes:
 n^n , $n \log(n)$; n^3 ; $\log(n)$, $n!$; n ; $k=cte$; e^n

ZONA DE RESPUESTAS