



Examen convocatoria de Junio de 2002

Modelo A

Nombre: _____

1. (2.5 puntos) Marca la respuesta verdadera de los conceptos que se señalan (Nota: los fallos descuentan puntos)

En la Unidad Central de Proceso (CPU) de un ordenador, los registros:

- ☐ Son utilizados para almacenar datos y constituyen una ampliación notable de la memoria del ordenador.
- ☐ Sirven para almacenar el programa en ejecución. Para que un programa se ejecute, éste debe cargarse primero en su totalidad en los registros de la CPU.
- ☐ sólo son accesibles por la Unidad de Control (UC). La Unidad Aritmetico-lógica (ALU) no puede acceder a éstos.
- ☐ almacenan unos pocos datos, pero muy importantes algunos de ellos. por ejemplo el Registro de Instrucciones contiene la siguiente instrucción del programa a ejecutar por la CPU.

Uno de los principios de la programación estructurada dice que:

- ☐ Existen tres tipos de sentencias iterativas: las de tipo `for`, las de tipo `while`, y las `do..while`
- ☐ Existen tres tipos de sentencias condicionales: las de tipo `if..else`, las anidadas `if..else if` y las múltiples (`switch`).
- ☐ Existen tres tipos de sentencias con las que se construyen programas: las iterativas, las condicionales y los enunciados.
- ☐ Hay tres tipos de metodologías para hacer programas: las descendente, la ascendente y la estática.

El concepto de ámbito de una variable:

- ☐ indica el tipo de dato que almacena la variable
- ☐ indica si dicha variable es un argumento del subprograma y si esta pasado por valor o por referencia
- ☐ indica el trozo de código del programa donde se puede utilizar dicha variable
- ☐ indica el rango de valores que admite dicha variable

Sea `int A` una variable declarada en `main()` y pasada *por referencia* a un subprograma. Si la variable se inicializó a `A=3` en `main()` y dentro del subprograma se ejecuta la instrucción `*A = 5` :

- ☐ Tras la llamada a la función, el valor de `A` sigue siendo 3
- ☐ Tras la llamada a la función, el valor de `A` es 5;
- ☐ Tras la llamada a la función, el valor de `A` es indefinido porque la asignación `*A =5`; es incorrecta. Sería `A=5`;
- ☐ La variable `A` es local al subprograma y desaparece cuando su llamada termina.



De las opciones a continuación, la manera más natural de recorrer una cadena de caracteres es:

- ☐ Usando un bucle `do . . while`
- ☐ Usando un bucle `for`
- ☐ Usando un condicional `if`
- ☐ Utilizando un doble bucle `for` anidado

Los argumentos de un subprograma

- ☐ Pueden ser sustituidos por el empleo de variables locales al subprograma.
- ☐ pueden ser formales o reales e incluso una mezcla de los dos
- ☐ son siempre tipos de datos simples
- ☐ No pueden ser omitidos ya que su papel es el de pasar datos del programa principal al subprograma y viceversa

Una diferencia entre los ficheros de tipo texto y los ficheros de tipo binario

- ☐ consiste en que los primeros pueden ser leídos por cualquier programa carácter a carácter, mientras que los segundos sólo pueden ser leídos correctamente si se conoce a priori los tipos de datos que almacena
- ☐ los primeros trabajan con el nivel lógico del ordenador y los segundos con el nivel físico
- ☐ en los primeros es imposible utilizar el acceso secuencial mientras que en los segundos sí se usa.
- ☐ En los primeros debo de comprobar si la apertura del fichero ha tenido éxito y en los segundos no.

Los vectores, matrices, cadenas y registros son tipos compuestos de datos porque:

- ☐ Sus componentes son accesibles a través de índices
- ☐ la composición de los datos determina su significado. Así por ejemplo en una cadena, el orden de sus componentes no es indiferente. No es lo mismo “Mero” que “Remo”.
- ☐ se pueden componer unos con otros, es decir, puedo componer un vector de cadenas por ejemplo
- ☐ son agregados de datos simples u otros compuestos, a los que podemos acceder individualmente de alguna forma.

Un parámetro formal en un subprograma

- ☐ Puede pasarse únicamente por valor.
- ☐ debe ser necesariamente de un tipo de dato simple.
- ☐ es el que utilizamos para codificar el código del subprograma del cual es parámetro.
- ☐ es la variable del programa que se pasa al subprograma en el momento de llamar al mismo.



En lenguaje C, la expresión $5/2$ da como valor numérico resultado

- ☐ 2.5
- ☐ Depende del tipo de dato de la variable a la que se le asigne
- ☐ 2
- ☐ Depende del compilador que utilicemos.

2. (2.5puntos) Encuentra los errores semánticos (de concepto, de uso) en los siguientes códigos en C. Señálalos con un número y pon en la hoja de respuestas el número y la explicación. Puede haber más de un error en cada cuadro.

CODIGO 1

```
int main(){
int i, aux, resul;
printf("introduce numero > 0\n");
scanf("%d", &aux);
for(i=0; i<aux; i--){
    resul = resul+ 1/ i;
printf("resultado: %d", resul);
}
```

CODIGO 2

```
int lee(){
FILE *fp;
int resul, dato;
resul = 0;
fp = fopen("datos.bin", "rb");
while(!feof(fp)){
    fread(&dato, sizeof(int), 1, fp);
    resul = resul + dato;
}
return(resul);
}
```

CODIGO 3

```
void ordenar(int a, int b, int c){
if(b < a){
    a = b; //intercambio valores
    b = a;
}
if(b < c){
    c = b; //intercambio valores
    b = c;
}
if(c < a){
    a = c; //intercambio valores
    c = a;
}

printf("%d < %d < %d \n", a,b,c);
}
```

CODIGO 4

```
void(sacamax(int *a){
    int aux;
    printf("Introduce numeros o el -1
para terminar");
    while(aux!=-1){
        scanf("%d", &aux);
        if(a < aux)
            a = aux;
    }
}

int main(){
int resul =0;
int max;
int i;

for(i=0;i<10;i++){
    sacamax(max);
    resul = resul + max;
}
printf("suma de maximos: %d", resul);
return(0);
}
```



3. (2.5 puntos)

a) Explica RAZONADAMENTE lo que hace el siguiente algoritmo: Nota: No me expliques

```
#include <stdio.h>
#define MAX 100

int esvalido(char c){
    if(c=='0' || c=='1' || c=='2' || c=='3' || c=='4' ||
       c=='5' || c=='6' || c=='7' || c=='8' || c=='9' ||
       c == '.')
        return (1);
    else
        return (0);
}

int main()
{
    int i;
    float factor;
    char cadena[MAX];
    float resul;
    char espunto;
```

```
printf("Introduce una cadena: ");
scanf("%s", cadena);
i=0;
resul = 0.0;
espunto='n';
factor = 1.0;
while((cadena[i]!='\0') && (esvalido(cadena[i])==1)){
    if(cadena[i]=='.'){
        espunto='s';
        factor = 10.0;
    }
    else if(espunto=='n'){
        resul = resul*factor + (cadena[i] - '0');
        factor = factor * 10.0;
    }
    else{
        resul = resul + (cadena[i] - '0') / factor;
        factor = factor * 10.0;
    }
    i++;
} //del while
if(cadena[i] == '\0')
    printf("\nla cadena es valida y su traduccion es:
           %f ", resul);
else
    printf("\nla cadena no es valida");
return 0;
} // cierra el main
```

lo que hace instrucción por instrucción.

b) Haz una traza del algoritmo, es decir indicando cómo se van transformando en función del valor de i la variable resul si la cadena de entrada es "3.1415";

c) ¿Qué ocurrirá si la cadena de entrada es "23.4ew1"?

4. (2.5 puntos)

a) Define los PROTOTIPOS (NO EL CODIGO) de los siguientes subprogramas de la manera que te parezca más razonable, prestando especial atención al decidir los pasos (por valor o referencia) de los argumentos (si es que tiene argumentos). Tras escribir el prototipo JUSTIFICA TU ELECCION con una breve explicación. Puedes poner el nombre que quieras al subprograma.

NOTA: sabes que la forma de un prototipo es:

<tipo de dato devuelto> <nombre subprograma> (<tipo argumento> <nombre argumento,...>);

a.1 Un subprograma que indique sacando por pantalla un mensaje cuál de los dos números reales recibidos es el mayor.

a.2 Un subprograma que reciba una cadena y devuelva cuántas vocales hay en ella

a.3 Un subprograma que reciba dos números enteros a y b y devuelva dos valores: el resultado de la división real de a entre b y el resultado real de la división de b entre a

a.4 Un subprograma que saque por pantalla las cinco letras vocales.

a.5 Un subprograma que tome una cadena de caracteres y devuelva la longitud de la cadena y el número de veces que aparece la letra 'a'.

b) Elabora el diagrama de flujo del siguiente código

```
a = 9;
while(a > 0){
    if(a==3){
        b=8;
        c = 9;
    }
    a --;
}
```