

NOMBRE Y APELLIDOS DEL ALUMNO:

## Informática

Prueba de conjunto

23.05.14

1.- **(0.25 puntos)** Considerar el siguiente fragmento de código:

```
for(i=0;i<5;i++)
{
    for(j=0;j<i;j++)
        printf("*");
    printf("\n");
}
```

Explicar qué se muestra en la pantalla cuando se ejecuta.

2.- **(0.25 puntos)** Considerar los siguientes fragmentos de código:

Fragmento A:

```
if(x<25) x++;
if(x>=25) x--;
```

Fragmento B:

```
if(x<25) x=x+1;
else x=x-1;
```

¿Son intercambiables? Si lo son, justificar la respuesta. Si no lo son, poner un ejemplo en el que funcionen de forma distinta.

3.- **(0.25 puntos)** Considerar el siguiente fragmento de código:

```
#define ESPACIO ' ' /*carácter espacio*/
#define TAB '\t' /*carácter tabulador*/
#define FINCADENA '\0' /*carácter fin de cadena*/
...
for(i=0,palabras=0;cadena[i]!=FINCADENA;)
{
    if( (cadena[i]=ESPACIO) || (cadena[i]=TAB) )
        palabras++;
    i++;
}
```

El fragmento pretende contar las palabras que hay en una cadena de texto, suponiendo que entre palabra y palabra sólo hay, o bien un único espacio, o bien un único carácter tabulador.

El fragmento no realiza la tarea pretendida. Indicar por qué.

4.- **(0.25 puntos)** Considerar el siguiente fragmento de código:

```
int longitud,i;
char cadena[100];

longitud=strlen(cadena);
for(i=(longitud-1);i>=0;i--)
{
    cadena[i]=cadena[longitud-i];
    cadena[longitud-i]=cadena[i];
}
```

El fragmento pretende invertir los caracteres de la cadena cadena. Sin embargo, no realiza esa tarea. Indicar por qué.

5.- (0.5 puntos) Considerar el siguiente programa:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAXALUMNOS 100

typedef struct
{
    char nombre[100];
    unsigned int edad;
    char dni[10];
    float nota;
}TAlumno;

int main()
{
    FILE *fichero;
    TAlumno alumno;
    int leido,buscado;
    float notamayor;

    fichero=fopen("notas.dat","r");
    leido=0;
    buscado=0;
    notamayor=0.0;
    while ( (fread(alumno,sizeof(TAlumno),1,fichero)!=0)
    {
        if(alumno.nota>notamayor)
        {
            notamayor=alumno.nota;
            buscado=leido;
            leido++;
        }
    }
    printf("\nAlumno buscado: %d",buscado);

    fclose(fichero);
}
```

El programa lee del fichero `notas.dat` una serie de registros que responden al formato del tipo `TAlumno`. El programa debería devolver, en la variable `buscado`, el orden que ocupa en el fichero el registro del alumno con la nota más alta. Sin embargo, no devuelve el valor correcto. Indicar por qué.

- 6.- (0.25 puntos) Se dispone de dos matrices de números en coma flotante, `matrizA` y `matrizB` cuyas dimensiones están almacenadas en las variables `filasA`, `columnasA`, `filasB` y `columnasB`. Se pide completar el siguiente fragmento de código con, como máximo, 2 sentencias de código en el recuadro para que calcule el producto de ambas, la matriz `producto`. Se supone que el producto es posible.

```
for (i=0; i<FilasA; i++)
    for(j=0; j<ColumnasB; j++)
    {
        suma=0.0;
```

```
        producto[i][j]=suma;
    }
```

- 7.- (0.25 puntos) Considerar el siguiente fragmento de código.

```
int filas,columnas;
float **matriz1,**matriz2;
int i,j;
```

...

```
matriz1=(float **)malloc(filas*sizeof(float *));
matriz2=(float **)malloc(columnas*sizeof(float *));
```

```
for(i=0;i<filas;i++)
    *(matriz1+i)=(float *)malloc(columnas*sizeof(float));
```

```
for(i=0;i<columnas;i++)
    *(matriz2+i)=(float *)malloc(filas*sizeof(float));
```

...

```
for(i=0;i<filas;i++)
    for(j=0;j<columnas;j++)
```

```
}
```

El fragmento debe transponer la `matriz1` y dejar el resultado en la `matriz2`. Rellenar el recuadro con, como máximo, 2 sentencias, para completar el fragmento.

8.- (0.5 puntos) Considerar el siguiente programa:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAXPALABRAS 100
#define MAXLETRAS 100

int main()
{
    char c, temp[MAXLETRAS];
    char **listapalabras;
    int i,j,numpalabras;

    listapalabras =(char **)malloc(MAXPALABRAS*sizeof(char *));

    j=0;
    numpalabras=0;

    printf("\nEscribir frase. Terminar con ENTER\n");
    do
    {
        c=getchar();
        if (c!=' ' && c!='\n')
            temp[j++]=c;
        else
        {
            temp[j++]='\0';
            listapalabras[numpalabras]=(char *)malloc(j*sizeof(char));
            strcpy(listapalabras[numpalabras],temp);
            j=0;
            numpalabras++;
        }
    }while (c!='\n');

    printf("\nFrase de salida: \n");
    for(i=0;i<numpalabras;i++)
    {
        for(j=strlen(listapalabras[i])-1;j>=0;j++)
            printf("%c", listapalabras[i][j]);
        printf(" ");
    }
}
```

El programa pretende leer una frase de entrada estándar y mostrarla en salida estándar invirtiendo el orden de los caracteres de sus palabras (no el orden de las palabras!). Ejemplo:

Frase de entrada: En un lugar de La Mancha

Frase de salida: nE nu ragul ed aL ahcnaM

Sin embargo, no lo hace, a consecuencia de un error. Indicar cuál.