

FUNDAMENTOS DE INFORMÁTICA

Curso 2007-2008

Ejercicios de repaso de programación en lenguaje C

Ejercicio 1.

Escribir la salida por pantalla del siguiente programa:

```
#include <stdio.h>
void main(void)
{
    int i,j;
    int a[5] = {2,4,6,10,12};
    int *p;
    i=5;
    while(i!=0)
    {
        i--;
        printf("%d %d %d\n", a[i]/3, i, a[i]);
    }
    p = a;
    for (j=0; j<=5; j++)
        printf("%d %d %d %d\n", a[i], *p, *(p+3), j);
    printf("%d\n", j);
}
```

Ejercicio 2.

Dado el siguiente programa en C, escribir la salida que se obtendría al ejecutarlo. Si en algún caso no puede determinarse el valor de alguna variable, escribir el símbolo ? en el lugar en el que debería imprimirse su valor.

```
#include <stdio.h>
#define TRUE 1
#define FALSE 0

short funcion1(int n)
{
    int i;

    i=n-1;
    while( (i>=2) && (n%i != 0) )
    {
        i = i-1;
    }
    if (i == 1)
        return (TRUE);
    else
        return (FALSE);
}

void funcion2(int *vector, int *vector2, int t)
{
    int i;
    for (i=t; i>=0; i--)
    {
        if (vector[i] > 0)
            vector2[t-i] = vector[i];
    }
}
```

```
void main(void)
{
    int i;
    int v1[10] = {-10,9,-8,7,-6,5,-4,3,-2,1};
    int v2[10];
    for (i=1; i<10; i++)
    {
        if (funcion1(i))
            printf("%d\n", i);
    }
    funcion2(v1,v2,10);
    printf("%d\n", v2[0]);
    printf("%d\n", v1[1]);
    printf("%d\n", v2[2]);
    printf("%d\n", v1[3]);
    printf("%d\n", v2[4]);
    printf("%d\n", v1[5]);
}
```

Ejercicio 3.

Escribir funciones en C que realicen las siguientes tareas:

- Una función llamada LeerVector que lea por teclado un vector de números enteros.
- Una función llamada Buscar que tome dos parámetros de entrada: un vector de enteros y un número entero, y que devuelva el índice de la primera aparición del número en el vector, o el valor -1 si el número no aparece en el vector.
- Una función llamada ImprimirPrimos que saque por pantalla todos los números del vector que sean primos
- Escribir una función main para comprobar el funcionamiento de los tres apartados anteriores.

Ejercicio 4.

Realizar un programa en C que recoja estos apartados:

- Escribir una función en C llamada Digitos que, dado como parámetro un número entero, imprima por pantalla el número con los dígitos impresos a la inversa y devuelva como resultado el número de dígitos que tiene el número. Por ejemplo, si el parámetro es 237, deberá imprimir 732 y devolver como resultado el número 3.
- Escribir una función en C llamada Cadenas que, dada una cadena de caracteres, devuelva un valor verdadero (1) si la cadena se lee igual de izquierda a derecha que de derecha a izquierda, y un valor falso (0) en caso contrario. Por ejemplo, si la cadena es "aba" la función debe devolver un valor verdadero y si la cadena es "abc" la función debe devolver un valor falso.
- Escribir un ejemplo de función main con una llamada a cada una de las funciones de los apartados anteriores.

Ejercicio 5.

Realizar un programa en C que:

- Lea una frase dada por el usuario y la introduzca en un vector.
- Busque si hay más de dos espacios vacíos entre las palabras y los quite para dejar sólo un espacio consecutivo.
- Imprima la frase resultado.

Por ejemplo, dada la frase:

Esto·es·una·frase·de·prueba

donde se han representado, sólo a modo ilustrativo, los espacios mediante puntos, la frase a obtener sería la siguiente:

Esto·es·una·frase·de·prueba

Ejercicio 6.

Un vector de N caracteres contiene el nombre completo de una persona, estando éste formado por uno o varios nombres y dos apellidos. Se pide un programa en C que devuelva en otro vector los apellidos de dicha persona y las iniciales de su(s) nombre(s) tal y como se indica a continuación:

El vector original contiene:

Jose Luis Garcia Pont (relleno a blancos hasta N)

Se debe obtener otro vector que contenga:

Garcia Pont, J.L.

Escribir el código necesario para leer e imprimir las cadenas, para comprobar el funcionamiento correcto del programa.

Nota 1: Una persona puede tener uno o más nombres.

Nota 2: Para simplificar el problema, suponer que no existen nombres ni apellidos del tipo "de ...", "de las ...", "de la ...", etc.

Ejercicio 7.

Un programa en C contiene las siguientes sentencias:

```
int A[10]={ 12,15,17,21,34,45};  
int *B;  
B=A;*B=A[4];
```

- ¿Qué valor representa A[4]?
- ¿Qué valor representa *(&B[3])?
- ¿Qué valor representa *(B+2)?
- ¿Qué valor representa A[0]?
- ¿Qué valor representa *A?
- ¿Qué valor representa *(B+0)?
- ¿Qué valor representa *(A+4)?
- ¿Qué valor representa &(A+4)?
- ¿Qué valor representa *A[3]?

Escribir un programa que imprima cada uno de los valores anteriores, comprobando que coinciden con lo contestado previamente.

Ejercicio 8.

Dado el siguiente programa en C, se pide:

- Escribir la salida que se obtiene en pantalla al ejecutarlo.
- Explicar qué resultado se obtiene en general en el parámetro r a partir del parámetro n.

```
#include <stdio.h>

#define BASE 10

int funcion(int n, int *r)
{
    int j;

    *r = 0;
    while(n >= BASE)
    {
        j = n % 10;
        n /= 10;
        *r += j;
    }
    *r += n;
    printf("%d %d %d\n", n, *r, j);
    if(n >= 0)
        return(j);
    else
        return(0);
}

void main(void)
{
    int a, b=0, c;

    a = 1725;
    c = 2;
    printf("%d %d %d\n", a, b, c);
    c = funcion(a, &b);
    printf("%d %d %d\n", a, b, c);
}
```

Ejercicio 9.

Con el siguiente programa en C, se pretende pedir al usuario que introduzca varios números (reservando memoria dinámicamente para almacenarlos) y a continuación calcular su media, varianza y desviación típica. Para ello, se ha creado una función llamada **Funcion1** que debe calcular y devolver estos tres valores. Se pide comprobar si existe algún error en el programa y en caso afirmativo, reescribir correctamente las sentencias erróneas.

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

void Funcion1(float *media, float *varianza, float *desviacion, float *vector, int n);

void main(void)
{
    int n, i;
    float *datos, med, var, desv;

    printf("Cuantos numeros va a introducir? ");
    scanf("%d", &n);
    datos = malloc(n*sizeof(float));
    if(datos == NULL)
        printf("\nError durante la reserva de memoria\n");
    else
    {
        for(i=0; i<n; i++)
        {
            printf("Introduzca el dato %d: ", i+1);
            scanf("%f", &(datos+i));
        }
        Funcion1(&med, &var, &desv, datos, n);
        printf("\nLa media es %f\nLa varianza es %f\nLa desviacion es %f\n", med, var, desv);
        free(datos);
    }

    system("PAUSE");
}

void Funcion1(float *media, float *varianza, float *desviacion, float *vector, int n)
{
    int i;

    for(i=0; i<=n; i++)
        *media += *(vector+i);
    *media /= n;

    for(i=0; i<n; i++)
        *varianza += (vector[i] - *media) * (vector[i] - *media);
    *varianza /= n;
    *desviacion = sqrt(*varianza);
}
```