Práctico 1

Programación en C

(I) Imprescindible
(R) Recomendado
(C) Complementario

Ejercicio 1 (I)

a) Considerando que las representaciones internas de int y float son 16 y 32 bits respectivamente. ¿Qué rango de valores pueden tomar las variables de los siguientes tipos?

`char – short – unsigned short – int – unsigned int – long – unsigned long - float – double`

b) Implemente un programa que imprima la cantidad de bits de cada uno de estos tipos de datos.

Ejercicio 2 (I)

Hay dos maneras de declarar constantes simbólicas en C:

```
#define <nombre> <valor>
const <tipo> <nombre> = <valor>
```

Discuta ventajas y desventajas de cada uno de estos métodos.

Ejercicio 3 (I)

Compile y ejecute los siguientes programas:

a)
```
#include <stdio.h>

int main ()
{
      cout << "Programa 1.\n" ;
      return 0;
}
```

b)
```
#include <stdio.h>

int main ()
{
      int n;
      cout <<"Ingrese un entero:";
      scanf ("%d", &n);
      cout << "Entero ingresado:" << n << endl;
      return 0;
}
```

EJERCICIO 4 (I)

Indique que imprimen los siguientes trozos de código:

a)
```
int a = 0;
if (a++)
  cout << "a es mayor que 0";
else
  cout << "a es 0";
```

b)
```
int a = 0;
if (++a)
  cout << "a es mayor que 0";
else
  cout << "a es 0";
```

EJERCICIO 5 (I)

Reescríba la siguiente función utilizando la proposición switch:

```
void printNota (int nota)
{
  if ((0 <= nota) && (nota < 3))
    cout << "Aplazado\n";
  else if ( (3 <= nota) && (nota < 7))
    cout <<"Aceptable\n";
  else if ( (7 <= nota) && (nota < 13))
    cout << "Bien\n";
  else
    cout << "Fuera de rango\n";
}
```

EJERCICIO 6 (I)

Dado el siguiente programa escrito en C:

```
#include <stdio.h>

#define MAXIMO_DE_LINEA 80

char* func (char* s)
{
    int i;
    for (i = strlen(s) - 1; i >= 0; i--)
        if ((s[i] <= 'z') && (s[i] >= 'a'))
            s[i] -= 'a' - 'A';
    return s;
}

int main ()
{
    char* s = (char*) malloc (sizeof (char) * MAXIMO_DE_LINEA);
    scanf ("%[^\n]", s);
    s = func(s);
    cout << s <<endl;
    return 0;
}
```

a) ¿Qué hace la función **`func`**?
b) ¿Que beneficios le trae la utilización del siguiente indicador de formato **"%[^\n]"**, al momento de leer de la entrada estándar?
c) ¿Que beneficios trae la manera de recorrer el arreglo (comenzando desde el final)?
d) ¿Que restricciones tiene este programa para que se ejecute correctamente?

EJERCICIO 7 (I)

Indique que efecto tiene cada uno de los siguientes trozos de código:

```
a) int c[5];
b) int c[5] = {0};
c) int n[5] = {10, 20, 30, 40, 50};
d) int n[] = {4, 3, 2, 1, 0};
e) char c[] = “hola”;
f) int d[3][3] = {{1, 2, 3}, {2, 3, 4}, {5, 6, 7}};
g) char c[2][2] = {48};
h) int t[3][3] = {{1,2,3}, {4}, {5,6}};
i) int a[3][3] = {1,2,3,4,5};
j) int a[3][3];
   for (int i = 0; i <= 2; i++)
     for (int j = 0; j <= 2; j++)
       a[i][j] = i + j;

k) char* c = “puntero_a_char”;

l) int c[3] = {‘1’,’2’,’3’};
   int *p = c;
   cout << *p;
   cout << p[0];
   cout << p[1];
   p++;
   cout << p[1];
```

m)
```
int *p = (int*) malloc (sizeof (int) * 4);
for (int i = 0; i <= 3; i++)
   p[i] = i;

int *p2 = p;
p2 += 3;
cout << *p2;
```

n)
```
int a = 1500;
int b = 2000;
int c = 3000;

int **doble_puntero;
doble_puntero = (int**) malloc (sizeof (int*) * 3);
doble_puntero[0] = &a;
doble_puntero[1] = &b;
doble_puntero[2] = &c;

for(int i = 0; i <= 2; i++)
  cout << *(doble_puntero[i]);
```

EJERCICIO 8 (I)

Explique cuáles son los errores de los siguientes trozos de código:

a)
```
int a[3];
a = new int[10];
```

b)
```
int a[3] = {1, 2, 3, 4};
```

c)
```
char* c = new char[10];
for (int i = 1; i <= 10; i++)
  c[i] = i;
```

d)
```
struct s{int a, b;};
s* puntero_a_s = new s;
*puntero_a_s.a = 1;
puntero_a_s ->b = 1;
(*puntero_a_s).a = 1;
```

e)
```
#define TAMANIO_MATRIZ = 10
int** matriz;
matriz = (int*) malloc (sizeof (int) * TAMANIO_MATRIZ);
```

f)
```
int numero = 5;
int* p_numero = numero;
```

g)
```
char* p_char = "pala";
p_char[2] = 't';
```

h)
```
int* func ()
{
      int i = 5;
      return &i;
}
int main ()
{
      int* p_i = func ();
      cout << *p_i;
      return 0;
}
```

i)
```
int a, b;

cout << "\nPor favor, ingrese un número: ";
cin >> a;
cout << "\nIngrese otro número: ";
cin >> b;

if (a = b) cout << "\n Son iguales";
else cout << "Son distintos";
```

j)
```
void darValor (int val, int* ptr)
{
      int* aux = new int;
      *aux = val;
      ptr = aux;
}

int main ()
{
      int* ptr;
      darValor (3, ptr);
      cout << *ptr;
      return 0;
}
```

EJERCICIO 9 (I)

Indicar errores (en caso de haberlos) y la salida de los siguiente programas:

a)

```
#include <stdio.h>

int main ()
{
      int i = 0, j = 0;
      for (i += 2; j < 6; j++, i += 2)
            cout << I << " " << j++ << " ";
      cout << endl;
      return 0;
}
```

b)

```
int main ()
{
    int a = 1, b = 0;
    int *p = &a;

    cout << " a = “<< a << endl;
    cout << " p = “<< p << endl;
    cout << "*p = “<< *p << endl;

    a = 2;
    cout << "a = “ << a << endl);
    cout << " p = “ << p << endl;
    cout << "*p = “ << *p << endl;

    *p += 2;
    cout << " a = “ << a << endl;
    cout << " p = " <<  p << endl;
    cout << "*p = " << *p << endl;

    p = &b;
    cout << " a = “ << a << endl;
    cout << " p = " <<  p << endl;
    cout << "*p = " << *p << endl;


    return 0;
}
```

c)

```
int main ()
{
    int a[10], i;
    int *p = a, *q = &(a[0]);

    for (i = 0; i < 10; i++) a[i] = i;

    cout << " p = " <<  p << endl;
    cout "q = “ << q << endl;

    for (i = 0; i < 10; i++, p++)
        printf("a[%d] = %d\n", i, *p);

    return 0;
}
```

d)

```
int main ()
{
    int i;
    char *s;

    s = new char[26];

    for (i = 0; i < 26; i++)
        s[i] = 'a' + i;

    for (i = 0; i < 26; i++)
        printf("s[%02d] = %c\n", i, s[i]);

    return 0;
```

```
}
```

EJERCICIO 10 (I)

Dada la siguiente función para intercambiar los valores de dos variables enteras:

```
void swap (int *a, int *b)
{
    int z = *a;
    *a = *b;
    *b = z;
}
```

Describa detalladamente el comportamiento del programa en cada caso.

a)

```
int main ()
{
    int *a, *b;
    swap(a, b);
    return 0;
}
```

b)

```
int main ()
{
    int a = 3, b = 5;
    swap(&a, &b);
    return 0;
}
```

c)

```
int main ()
{
    int a = 3, b = 5;
    swap(a, b);
    return 0;
}
```

d)

```
int main ()
{
    int a = 3, b = 5;
    int *p = &a, *q = &b;
    swap(p, q);
    return 0;
}
```

EJERCICIO 11 (I)

Indique el efecto de la llamada a la función `inicializar` en cada caso:

a)

```
void inicializar (int *a)
{
    a = new int;
    *a = 4;
}
```

```
int main ()
{
    int *a;
    inicializar(a);
    return 0;
}
```

b)

```
void inicializar (int **a)
{
    *a = new int;
    **a = 4;
}
```

i)

```
int main ()
{
    int **a;
    inicializar(a);
    return 0;
}
```

ii)

```
int main ()
{
    int *p;
    inicializar(&p);
    return 0;
}
```

EJERCICIO 12 (I)

Escriba un programa que calcule aproximaciones del valor de π a partir de las reducidas n-ésimas de la siguiente serie infinita:

$$\pi = 4 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + ...$$

EJERCICIO 13 (R)

a) Escriba una función que reciba un número natural como parámetro y devuelva su factorial. Discuta que tipos conviene utilizar para el n y para acumular el resultado. Observe que es posible que el n sea suficientemente grande como para que el resultado no sea representable. ¿Cómo resolvería este problema?
b) Escriba un programa que reciba un número de la línea de comandos y despliegue su factorial en la salida estándar. El programa debe lidiar correctamente con las situaciones de error posibles. Discuta ventajas y desventajas de las siguientes alternativas para el chequeo de precondiciones, e implementelas:
    i. el invocador chequea las precondiciones
    ii. la función chequea las precondiciones
c) ¿Conviene que la función emita un mensaje de error? ¿Donde desplegaría los mensajes de error?

EJERCICIO 14 (R)

Un triángulo rectángulo puede tener lados que sean todos enteros. El conjunto de tres valores enteros para los lados de un triángulo se conoce como una terna pitagórica. Estos tres lados deben satisfacer la relación de que la suma de los cuadrados de dos de los lados es igual al cuadrado de la hipotenusa. Encuentre todas las ternas pitagóricas para lado 1, lado 2 e hipotenusa todos ellos no mayores de 50. Utilice un ciclo `for` de triple anidamiento que pruebe todas las posibilidades.

EJERCICIO 15 (C)

Escriba una función "multiplo" que determine para un par de enteros si el segundo de ellos es múltiplo del primero. La función debe tomar dos argumentos enteros y regresar verdadero si el segundo es un múltiplo del primero y falso de no ser así. Utilice esta función en un programa que introduzca una serie de pares de enteros.

EJERCICIO 16 (C)

Un número entero se dice un número perfecto si sus divisores, incluyendo a 1 (pero excluyendo al número mismo) suman igual que el número. Por ejemplo 6 es un número perfecto porque 6=1+2+3. Escriba una función "perfecto" que determine si el parámetro "n" es un número perfecto. Utilice esta función en un programa que determine e imprima todos los números perfectos entre 1 y 1000. Imprima los factores de cada número perfecto para poder confirmar que el número es efectivamente perfecto.

EJERCICIO 17 (C)

Escriba un programa que calcule la descomposición en factores primos de un número dado. El programa debe devolver que primos componen un número y con que multiplicidad.

EJERCICIO 18 (I)

Un palíndromo es una cadena que se escribe de la misma forma hacia adelante y hacia atrás. Algunos ejemplos de palíndromos son "radar", "able was i ere i saw elba" y "a man a plan a canal panama". Escriba una función recursiva "testPalindromo" que devuelva true si la cadena almacenada en un arreglo es un palíndromo y false de lo contrario. La función debe ignorar espacios y puntuaciones incluidas en la cadena.

EJERCICIO 19 (R)

Escriba una función que dado un número n, calcule el n-ésimo número de Fibonacci. Los números de Fibonacci se definen como

$$fib(0) = 0$$
$$fib(1) = 1$$
$$fib(n) = fib(n-1) + fib(n-2) \quad \text{si n} > 1$$

EJERCICIO 20 (C)

El máximo común divisor entre dos naturales *x* e *y* puede calcularse según:

Si y = 0 $mcd(x, y) = x$
Si no $mcd(x, y) = mcd(y, x \bmod y)$

Dos número naturales son primos entre sí, si su máximo común divisor es 1. Desarrolle un programa que dados dos naturales, determine si son primos entre sí.

EJERCICIO 21 (I)

Defina e implemente los siguientes TAD:

a) Lista de enteros
b) Pila de enterosa
c) Cola de enteros

Utilizando memoria estática (arreglos) y memoria dinámica (listas encadenadas).
Notar que son 2 implementaciones *distintas* por cada TAD: acotada y no acotada.