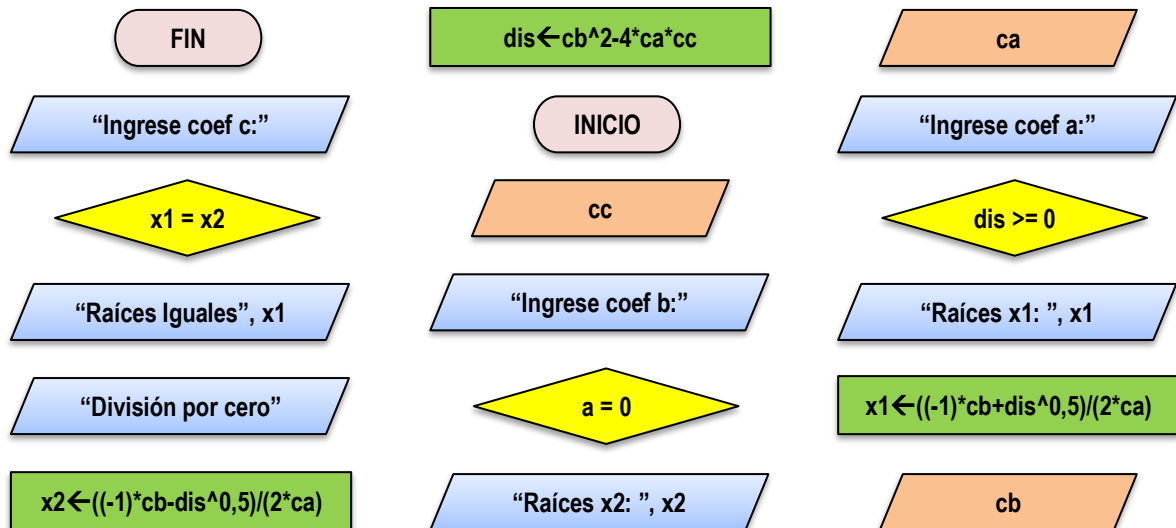
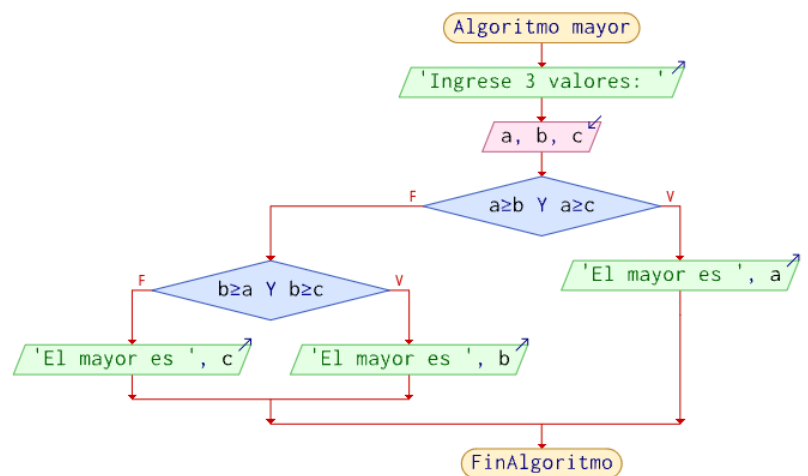


EJERCICIOS

- 1) Sabiendo que los siguientes bloques corresponden a un algoritmo (DIAGRAMA DE FLUJO) que calcula las raíces de una ecuación cuadrática, conéctelos en el orden correcto.



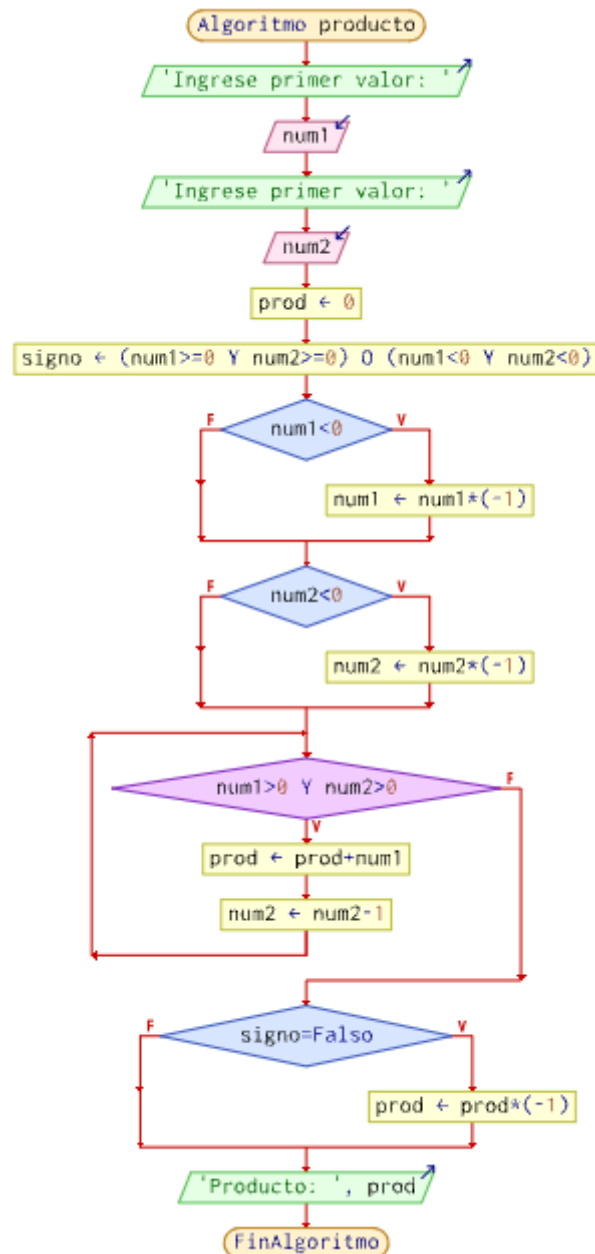
- 2) El siguiente diagrama de flujo corresponde a un algoritmo que identifica el mayor de 3 valores ingresados por el usuario. A partir de éste diseñe las versiones que se indican a continuación:



- se deben incluirse controles para verificar que los 3 valores sean distintos, caso contrario, deben presentarse los mensajes “3 valores iguales” o “2 valores iguales” según corresponda.
 - a partir del ítem a), debe modificarse el algoritmo para mostrar los valores en orden creciente.
 - a partir del ítem b), debe incluirse un bucle para repetir el ingreso de datos, en tanto los 3 valores no sean cero.
- 3) Diseñe un algoritmo (DIAGRAMA DE FLUJO) que calcule el cociente, mediante restas sucesivas, de 2 números enteros ingresados por el usuario. Considere que los valores de entrada pueden ser positivos, negativos o cero por lo que debe incluir los controles necesarios para realizar correctamente el cálculo. Realice la prueba de escritorio para A=13, B=-4 y A=-12, B=-5. Codifique en C/C++ el algoritmo diseñado.

4) El siguiente diagrama de flujo corresponde a un algoritmo que calcula el producto de 2 números enteros ingresados por el usuario. A partir de éste diseñe las versiones que se indican a continuación:

- el bucle de cálculo debe implementarse con el criterio de finalización por BANDERA.
- el bucle de cálculo debe ejecutar el menor número de repeticiones posible. Por ejemplo, si $num1=5$ y $num2=3$, el bucle debe realizar 3 repeticiones; mientras que si $num1=4$ y $num2=7$, el bucle debe ejecutar 4 iteraciones.
- realice la prueba de escritorio del ítem b) para $num1=-3$ y $num2=7$.



5) Realice la prueba de escritorio de los siguientes módulos, determine su propósito y escriba el código C/C++ o pseudocódigo equivalente según corresponda.

```

FUNCIÓN misterio (E n: ENTERO): ENTERO
VARIABLE
  b: LÓGICO
  s, x: ENTERO
INICIO
  SI n < 0 ENTONCES
    n ← n * (-1)
  FIN_SI
  x ← n
  s ← 0
  b ← x > 0
  MIENTRAS b = VERDADERO HACER
    s ← s + x * 2
    x ← x - 1
    SI x <= 0 ENTONCES
      b ← FALSO
    FIN_SI
  FIN_MIENTRAS
  misterio ← s - n
FIN
  
```

```

float enigma(int a, int b)
{ float p=1;
  bool neg=false;
  if (b<0)
  { neg=true;
    b=b*(-1);
  }
  while (b>0)
  { p=p*a;
    b--;
  }
  if (neg==true)
    p=1/p;
  return p;
}
  
```

- 6) Sabiendo que el siguiente programa implementa una calculadora sencilla, reemplace los módulos de cálculo por sus correspondientes versiones recursivas.

```
#include <iostream>

using namespace std;

void ingreso1(int &a);
void ingreso2(int &a,int &b);
int producto(int a,int b);
int cociente(int a,int b);
int factorial(int a);
int fibo(int a);

main()
{ int num1,num2,op;
  do {
    system("cls");
    cout << "1-Producto" << endl;
    cout << "2-Cociente" << endl;
    cout << "3-Factorial" << endl;
    cout << "4-Fibonacci" << endl;
    cout << "5-Salir" << endl;
    cout << "Ingrese opcion: ";
    cin >> op;
    switch(op)
    { case 1: ingreso2(num1,num2);
      cout << "Producto: ";
      cout << producto(num1,num2) << endl;
      break;
      case 2: ingreso2(num1,num2);
      cout << "Cociente: ";
      cout << cociente(num1,num2) << endl;
      break;
      case 3: ingreso1(num1);
      cout << "Factorial: ";
      cout << factorial(num1) << endl;
      break;
      case 4: ingreso1(num1);
      cout << "Fibonacci: ";
      cout << fibo(num1) << endl;
      break;
      case 5: cout << "The End" << endl;
      break;
      default: cout << " Incorrecto" << endl;
    }
    system("pause");
  } while(op!=5);
}
```

```
void ingreso1(int &a)
{ do
  { cout << "Ingrese valor: ";
    cin >> a;
    if (a<0)
      cout << "Ingrese positivos" << endl;
  } while(a<=0);
}

void ingreso2(int &a,int &b)
{ do
  {cout << "Ingrese valor: ";
    cin >> a;
    cout << "Ingrese valor: ";
    cin >> b;
    if (a<0 || b<=0)
      cout << "Ingrese positivos" << endl;
  } while(a<0 || b<=0);
}

int producto(int a,int b)
{ int s=0;
  while (b>0)
  { s=s+a;
    b--; }
  return s;
}

int cociente(int a,int b)
{ int c=0;
  while (a-b>0)
  { a=a-b;
    c++; }
  return c;
}

int factorial(int a)
{ int p=1;
  while (a>0)
  { p=p*a;
    a--; }
  return p;
}

int fibo(int a)
{ int x=1,y=1,z=1;
  while (a>2)
  { z=x+y;
    x=y;
    y=z;
    a--; }
  return z;
}
```

- 7) Usando la estructura general de programa, diseñe un programa modular que permita manejar un vector de enteros de 50 elementos. Considere que el programa debe presentar un menú con las siguientes opciones: 1) Agregar un valor al vector, 2) Buscar un valor en el vector, 3) Determinar los valores máximo y mínimo del vector, y 4) Salir, que presente el mensaje "Fin de programa". Cada opción debe implementarse mediante módulos especificando el paso de parámetros.

Nota: recuerde que las operaciones de inserción/eliminación sobre arreglos sólo insertan/eliminan un valor cuando son invocadas, NO realizan cargas completas o eliminaciones totales.

- 8) Dado un vector de caracteres de 30 elementos:
- Consigne la declaración de tipos y variables de la estructura.
 - Diseñe un procedimiento/función que permita insertar valores en el vector, considerando un criterio decreciente.
 - Diseñe un procedimiento/función que determine si un valor se encuentra o no el vector. Considere que la búsqueda será secuencial y que el algoritmo debe plantearse de modo que se aproveche que los datos están ordenados.
 - Diseñe un procedimiento/función **recursivo** que muestre el contenido del vector desde las primeras posiciones hacia las últimas o viceversa según un parámetro de opción.
- 9) Dada una matriz 9x9 de caracteres:
- Consigne la declaración de tipos y variables de la estructura.
 - Diseñe un procedimiento/función que cargue la matriz con símbolos alfabéticos (mayúsculas y/o minúsculas) aleatorios.
 - Diseñe un procedimiento/función que identifique el máximo valor de la diagonal principal de la matriz.
 - Diseñe un procedimiento/función que muestre, por cada COLUMNA, el menor valor almacenado.
- 10) Dada una matriz 5x7 de enteros:
- Consigne la declaración de tipos y variables de la estructura.
 - Diseñe un procedimiento/función que cargue la matriz con valores aleatorios del intervalo [1, 9999].
 - Diseñe un procedimiento/función que reemplace, cada valor de la matriz, por la cantidad de dígitos que le corresponde. Considere que la cantidad de dígitos debe determinarse mediante un módulo **recursivo** basado en el método de descomposición de dígitos.
 - Diseñe un procedimiento/función que determine si la matriz contiene exclusivamente valores impares.
- 11) Considerando el siguiente vector:

8	23	7	27	4	18	11
---	----	---	----	---	----	----

- Indique, gráficamente y paso a paso, cómo se realiza la ordenación **creciente** del vector aplicando el *método de ordenación por Selección*.
 - Sabiendo que, en cada recorrido, el algoritmo de ordenación *Burbuja* compara posiciones adyacentes del arreglo, realizando intercambios cuando sea necesario; modifique este algoritmo de modo que tras cada recorrido se omitan las posiciones ya ordenadas. Considere criterio de ordenación creciente.
- 12) El director del departamento de Recursos Humanos de una empresa desea registrar información acerca de sus 2300 empleados. Por cada empleado debe registrarse: legajo, apellido, nombre, dni, fecha de nacimiento (día, mes, año), domicilio (calle, número, barrio, localidad), cargo, fecha de ingreso (día, mes, año) sueldo básico y horas extra. Para ello, se solicita:
- Consigne la declaración de tipos y variables de la estructura que represente la situación planteada.
 - Diseñe los procedimientos/funciones que **listen** los empleados (apellido, nombre y año de ingreso) que no hayan realizado horas extra. Indique cuántos empleados se listaron.
 - Diseñe los procedimientos/funciones que **muestren** los empleados (legajo, cargo y sueldo básico) que pertenezcan a una localidad especificada por el usuario.
- 13) Dado el siguiente módulo, ¿cuál es la definición de datos que le corresponde?

```
void listar_alumnos(tvector alumnos, int ocup, tcad materia)
{int i, cont=0;
  for(i=0;i<=ocup;i++)
    if (strcmp(alumnos[i].materia,materia)==0)
    { cout << "Libreta: " << alumnos[i].lu << endl;
      cout << "Apellido: " << alumnos[i].apellido << endl;
      cout << "Nombre: " << alumnos[i].nombre << endl;
      cout << "DNI: " << alumnos[i].dni << endl;
      cout << "Carrera: " << alumnos[i].carrera.nombre << endl;
      cout << "Plan de estudios: " << alumnos[i].carrera.plan << endl;
      cout << "Promedio: " << alumnos[i].promedio << endl;
      cout << "Año de cursado: " << alumnos[i].inscripcion.año;
    }
}
```

- 14) El gerente de una clínica veterinaria desea registrar información acerca de las 200 mascotas que allí se atienden. Por cada mascota se debe guardar: id de mascota, nombre, edad, tipo (especie, raza, color) y dueño (apellido, nombre, domicilio, teléfono). Para ello, se solicita:
- Consigne la declaración de tipos y variables de la estructura que represente la situación planteada.
 - Diseñe los procedimientos/funciones que **listen** las mascotas que pertenezcan a una especie y raza solicitada por el usuario.
 - Diseñe los procedimientos/funciones que permitan **ordenar** (en forma creciente) las mascotas por especie.
- 15) El presidente de la asociación de fútbol de Jujuy desea informatizar el registro de información de los jugadores y equipos que desarrollan sus actividades en la provincia. Por cada jugador se debe guardar: id de jugador, apellido, nombre, apodo, fecha de nacimiento (día, mes, año), contacto (email, teléfono), domicilio (calle, número, barrio y localidad) e id de equipo. Y por cada equipo se debe guardar: id de equipo, nombre de equipo, director técnico (apellido, nombre y título profesional), fecha de fundación (día, mes, año), cantidad de campeonatos jugados y cantidad de campeonatos ganados. Para ello, se solicita:
- Consigne la declaración de tipos y variables de la estructura que represente la situación planteada.
 - Diseñe los procedimientos/funciones que **listen**, para cada equipo (nombre del equipo y nombre del director técnico), sus jugadores (apellido, nombre y localidad de residencia). Indique la cantidad de jugadores registrados por cada equipo.
 - Diseñe los procedimientos/funciones que **muestre** el equipo (nombre del equipo y cantidad de campeonatos jugados) con la mayor cantidad de campeonatos ganados.

