CI3641 - Lenguajes de Programación I

Examen 2 — *25 puntos*

1. Seleccioné el lenguaje **Fortran**.

   (a) Descripción del Lenguaje:

   Fortran posee 5 tipos de datos simples: `INTEGER`, `REAL`, `COMPLEX`, `LOGICAL` y `CHARACTER`.

   Para los `INTEGER`, se puede definir el tipo de entero que se quiere, y los números estarán en el rango $\pm 10^{<tipo>}$. Adicionalmente, se pueden definir la base en la que se quiere que se muestren los números (binaria, decimal, octal, hezadecimal).

   Los `COMPLEX` son tratados como dos `INTEGER` o dos `REAL` que representan la parte real y la parte imaginaria del número complejo.

   Se pueden definir tipos de datos compuestos a través de la palabra TYPE, que acepta que se definan atributo de cualquiera de los tipos previamente definidos (puede ser otro tipo de datos compuesto).

   También existen los arreglos, que son colecciones del mismo tipo de datos con determinadas dimensiones. Estos arreglos pueden ser dinámicos o estáticos, ya que se puede cambiar su tamaño si no se declaran con valores fijos, a través de las palabras claves `allocate` y `deallocate`, para reservar y liberar memoria respectivamente.

   Finalmente, en Fortran existe la noción de punteros, los cuales pueden hacer referencia a cualquiera de los tipos de datos previamente mencionados.

   Fotran no provee un mecanismo para la comparación de tipos derivados (`TYPE`), si es requerido hacer comparaciones habrá que sobrecargar el operador. Si una variable no ha sido previamente declarada, Fortran asume el tipo del valor resultante en el r-value. Si una variable que no es declarada comienza por las letras I, J, K, L, M, o N, se castea implicitamente como un `INTEGER`. Todas las demás se castearán como un `REAL`.

   Existen funciones de *casteo* para todos los tipos simples, por lo que podremos pasar de un `INTEGER` a `REAL` o del `CHARACTER` `".True."` al `LOGICAL` `.True.`.

   Fortran hace la distinción entre funciones y subrutinas, donde el primero devuelve un valor y el segundo no. Los parámetros son pasados por valor, por lo que al terminar la ejecución de una función o subrutina puede que los parámetros se encuentre modificados.

   No se maneja la noción de Excepción, por lo que no existen mecanismos para ello. Sin embargo, si se deseara utilizar, se podría generar una función que

retorne un número entero que tenga algún valor asociado (esta sería el tipo de excepción) para que posteriormente otra rutina se encargue de gestionarla.

(b) Las implementaciones las puedes encontrar en: Repositorio de Github :)

2. Las variables con mi número de carné quedaría de la siguiente manera:

```
int tu = 10, que = 9;

proc lagarto(int x, int y, int z) {
    x = x + y - z
    y = z * 2 - x
    z = x + y - 1
}

largarto(tu, tu, que)
largarto(tu, que, que)
largarto(que, tu, tu)

print(tu, que)
```

La respuesta a esta pregunta la realicé con la plantilla que usé en el primer parcial, por lo que va en un PDF en el Repositorio de Github :)

3. Como refiere a la implementación de un algoritmo, la puedes encontrar en: Repositorio de Github :)

4. Las variables con mi número de carné quedarían de la siguiente manera:

$A = 2 * ((4 + 6)\ mod\ 5) + 3$
$B = 2 * ((6 + 3)\ mod\ 5) + 3$
$A = 2 * 3 = 6 = A$
$B = 2 * 4 + 3 = 8 + 3 = 11 = B$

Y obtendríamos las siguientes corrutinas:

```
coroutine w():
  int a = 0
  loop:
    a = a + 6
    print(a)
    if a mod 2 == 0:
      transfer t()
    else:
      transfer f()
```

```
coroutine t():
  int b = 1
  loop:
    b = (b + 1) * 11
    print(b)
    if b mod 2 == 0:
      transfer w()
    else:
```

```
9       transfer f()
```

```
1  coroutine f():
2    int c = 1
3    loop:
4      c = c + 1
5      print(c)
6      if c mod 2 == 0:
7        transfer w()
8      else:
9        transfer t()
```

Solución:

PASO 1:

```
/* coroutine w */
pc -> 1
a -> No Empilada
ACTIVE
```

```
/* coroutine t */
pc -> No Iniciado
b -> No empilada
DEACTIVE
```

```
/* coroutine f */
pc -> No Iniciado
c -> No empilada
DEACTIVE
```

PASO 2:

```
/* coroutine w */
pc -> 2
a -> 0
ACTIVE
```

```
/* coroutine t */
pc -> No Iniciado
b -> No empilada
DEACTIVE
```

```
/* coroutine f */
pc -> No Iniciado
c -> No empilada
DEACTIVE
```

PASO 3:

```
/* coroutine w */
pc -> 4
a -> 6
ACTIVE
```

```
/* coroutine t */
pc -> No Iniciado
b -> No empilada
DEACTIVE
```

```
/* coroutine f */
pc -> No Iniciado
c -> No empilada
DEACTIVE
```

PASO 4:

```
/* coroutine w */
pc -> 5
a -> 6
ACTIVE
```

```
/* coroutine t */
pc -> No Iniciado
b -> No empilada
DEACTIVE
```

```
/* coroutine f */
pc -> No Iniciado
c -> No empilada
DEACTIVE
```

IMPRIME: 6

PASO 5:

```
/* coroutine w */
pc -> 7
```

```
a -> 6
DEACTIVE
```

```
/* coroutine t */
pc -> 1
```

```
b -> No empilada        /* coroutine f */       c -> No empilada
ACTIVE                  pc -> No Iniciado       DEACTIVE
```

PASO 6:

```
/* coroutine w */       /* coroutine t */       /* coroutine f */
pc -> 7                 pc -> 2                 pc -> No Iniciado
a -> 6                  b -> 1                  c -> No empilada
DEACTIVE                ACTIVE                  DEACTIVE
```

PASO 7:

```
/* coroutine w */       /* coroutine t */       /* coroutine f */
pc -> 7                 pc -> 4                 pc -> No Iniciado
a -> 6                  b -> 22                 c -> No empilada
DEACTIVE                ACTIVE                  DEACTIVE
```

PASO 8:

```
/* coroutine w */       /* coroutine t */       /* coroutine f */
pc -> 7                 pc -> 5                 pc -> No Iniciado
a -> 6                  b -> 22                 c -> No empilada
DEACTIVE                ACTIVE                  DEACTIVE
```

IMPRIME: 22

PASO 9:

```
/* coroutine w */       /* coroutine t */       /* coroutine f */
pc -> 7                 pc -> 7                 pc -> No Iniciado
a -> 6                  b -> 22                 c -> No empilada
ACTIVE                  DEACTIVE                DEACTIVE
```

PASO 10:

```
/* coroutine w */       /* coroutine t */       /* coroutine f */
pc -> 4                 pc -> 7                 pc -> No Iniciado
a -> 12                 b -> 22                 c -> No empilada
ACTIVE                  DEACTIVE                DEACTIVE
```

PASO 11:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 5               pc -> 7               pc -> No Iniciado
a -> 12               b -> 22               c -> No empilada
ACTIVE                DEACTIVE              DEACTIVE
```

IMPRIME: 12

PASO 12:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7               pc -> 7               pc -> No Iniciado
a -> 12               b -> 22               c -> No empilada
DEACTIVE              ACTIVE                DEACTIVE
```

PASO 13:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7               pc -> 4               pc -> No Iniciado
a -> 12               b -> 253              c -> No empilada
DEACTIVE              ACTIVE                DEACTIVE
```

PASO 14:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7               pc -> 5               pc -> No Iniciado
a -> 12               b -> 253              c -> No empilada
DEACTIVE              ACTIVE                DEACTIVE
```

IMPRIME: 253

PASO 15:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7               pc -> 9               pc -> 1
a -> 12               b -> 253              c -> No empilada
DEACTIVE              DEACTIVE              ACTIVE
```

PASO 16:

```
/* coroutine w */      a -> 12               /* coroutine t */
pc -> 7               DEACTIVE              pc -> 9
```

```
b -> 253
DEACTIVE
```

```
/* coroutine f */
pc -> 2
```

```
c -> 1
ACTIVE
```

PASO 17:

```
/* coroutine w */
pc -> 7
a -> 12
DEACTIVE
```

```
/* coroutine t */
pc -> 9
b -> 253
DEACTIVE
```

```
/* coroutine f */
pc -> 4
c -> 2
ACTIVE
```

PASO 18:

```
/* coroutine w */
pc -> 7
a -> 12
DEACTIVE
```

```
/* coroutine t */
pc -> 9
b -> 253
DEACTIVE
```

```
/* coroutine f */
pc -> 5
c -> 2
ACTIVE
```

IMPRIME: 2

PASO 19:

```
/* coroutine w */
pc -> 7
a -> 12
ACTIVE
```

```
/* coroutine t */
pc -> 9
b -> 253
DEACTIVE
```

```
/* coroutine f */
pc -> 7
c -> 2
DEACTIVE
```

PASO 20:

```
/* coroutine w */
pc -> 4
a -> 18
ACTIVE
```

```
/* coroutine t */
pc -> 9
b -> 253
DEACTIVE
```

```
/* coroutine f */
pc -> 7
c -> 2
DEACTIVE
```

PASO 21:

```
/* coroutine w */
pc -> 5
a -> 18
ACTIVE
```

```
/* coroutine t */
pc -> 9
b -> 253
DEACTIVE
```

```
/* coroutine f */
pc -> 7
c -> 2
DEACTIVE
```

IMPRIME: 18

PASO 22:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 9                pc -> 7
a -> 18                b -> 253               c -> 2
DEACTIVE               ACTIVE                 DEACTIVE
```

PASO 23:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 4                pc -> 7
a -> 18                b -> 2794              c -> 2
DEACTIVE               ACTIVE                 DEACTIVE
```

PASO 24:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 5                pc -> 7
a -> 18                b -> 2794              c -> 2
DEACTIVE               ACTIVE                 DEACTIVE
```

IMPRIME: 2794

PASO 25:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 7                pc -> 7
a -> 18                b -> 2794              c -> 2
ACTIVE                 DEACTIVE               DEACTIVE
```

PASO 26:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 4                pc -> 7                pc -> 7
a -> 24                b -> 2794              c -> 2
ACTIVE                 DEACTIVE               DEACTIVE
```

PASO 27:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 5                pc -> 7                pc -> 7
a -> 24                b -> 2794              c -> 2
ACTIVE                 DEACTIVE               DEACTIVE
```

IMPRIME: 24

PASO 28:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 7                pc -> 7
a -> 24                b -> 2794              c -> 2
DEACTIVE               ACTIVE                 DEACTIVE
```

PASO 29:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 4                pc -> 7
a -> 24                b -> 30745             c -> 2
DEACTIVE               ACTIVE                 DEACTIVE
```

PASO 30:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 5                pc -> 7
a -> 24                b -> 30745             c -> 2
DEACTIVE               ACTIVE                 DEACTIVE
```

IMPRIME: 30745

PASO 31:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 9                pc -> 7
a -> 24                b -> 30745             c -> 2
DEACTIVE               DEACTIVE               ACTIVE
```

PASO 32:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 9                pc -> 4
a -> 24                b -> 30745             c -> 3
DEACTIVE               DEACTIVE               ACTIVE
```

PASO 32:

```
/* coroutine w */      /* coroutine t */      /* coroutine f */
pc -> 7                pc -> 9                pc -> 5
a -> 24                b -> 30745             c -> 3
DEACTIVE               DEACTIVE               ACTIVE
```

IMPRIME: 3

5. Puedes encontrar la implementación en: Repositorio de Github :)