UNIVERSIDAD NACIONAL DE INGENIERÍA

CENTRO DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

CTIC UNI

C/C++ PROGRAMMING LANGUAGE
TEMA 4: OPERATORS, EXPRESSIONS, AND STATEMENTS

Nombres y apellidos: _____

Nombres y apellidos del instructor: MSc. César Manuel Sebastián Díez Chirinos.

1. Asuma que todas las variables son del tipo `int`. Encuentre el valor de cada una de las siguientes variables:
   (a) `x = (2 + 3) * 6;`
   (b) `x = (12 + 6)/2*3;`
   (c) `y = x = (2 + 3)/4;`
   (d) `y = 3 + 2*(x = 7/2);`

Solución

Si tenemos el siguiente listing 1, obtendremos para los valores de x e y:

```c
#include <stdio.h>
int main(void)
{
    int x, y;

    x = (2 + 3) * 6;
    printf("x: %d\n", x);

    x = (12 + 6)/2*3;
    printf("x: %d\n", x);

    y = x = (2 + 3)/4;
    printf("x: %d\ty: %d\n", x, y);

    y = 3 + 2*(x = 7/2);
    printf("x: %d\ty: %d\n", x, y);

    return 0;
}
```

Listado 1: Programa `exercise4_1.c`.

(a) Imprimirá:

```
pc@CTIC:~$ x: 30
```

(b) Imprimirá:

```
pc@CTIC:~$ x: 27
```

(c) Imprimirá:

```
pc@CTIC:~$ x: 1    y: 1
```

(d) Imprimirá:

```
pc@CTIC:~$ x: 3    y: 9
```

2. Sospechas que hay algunos errores en el siguiente programa. ¿Puede encontrarlos?

```c
int main(void)
{
        int=1,
        float n;
```

```
5           printf("watch out! Here come a bunch of fractions!\n");
6           while(i<30)
7                   n=1/i;
8                   printf(" %f", n);
9
10  printf("That's all, folks!\n");
11  return;
12  }
```

Solución

Línea 0: Debe ir la cabecera `#include <stdio.h>`.

Línea 3: Debe acabar en punto y coma, no en coma.

Línea 6: La instrucción **`while`** establece un ciclo infinito porque el valor de `i` permanece `1` y siempre es menos de `30`. Presumiblemente, se quiso escribir **`while`** `(i++ <30)`.

Línea 6–8 La sangría implica que queríamos las líneas 7 y 8 para formar un bloque, pero la falta de llaves significa que el ciclo while incluye solo la línea 7. Deben agregarse llaves.

Línea 7: Como `1` e `i` son ambos enteros, el resultado de la división será `1` cuando `i` es `1` y `0` para todos los valores más grandes. La sentencia `n = 1.0 / i;` causaría que `n` se convierta en flotante.

Línea 8: Omitimos un carácter de nueva línea `\n` en la declaración de control. Esto causa los números sean impresos en una sola línea.

Línea 10: Debería estar **`return`** `0;`.

Aquí hay una posible versión correcta:

```
#include <stdio.h>
int main(void)
{
        int i = 1;
        float n;
        printf("watch out! Here come a bunch of fractions!\n");
        while(i++ <30)
        {
                n = 1.0/i;
                printf(" %f\n", n);
        }
        printf("That's all, folks!\n");
        return 0;
}
```

Listado 2: Programa `exercise4_2.c`.

3. Hacer un `min_sec` interactivo no es fácil. ¿Cómo se puede mejorar?

```
#include<stdio.h>
#define S_TO_M 60
main()
{
int sec, min, left;
printf("This program convierte segundos a minutos y");
printf("segundos.\n");
printf("Solo debe ingresar los segundos.\n");
printf("Ingrese 0 para finalizar el programa.\n");
while(sec>0){
scanf("%d", &sec);
```

```
min=sec/S_TO_M;
left=sec%S_TO_M;
printf("%d sec is %d min, %d sec.\n", sec, min, left);
printf("Next input?\n");
}
printf("See you!\n");
}
```

Solución

El principal problema radica en la relación entre la declaración de prueba (¿es `sec` mayor que 0?) Y la instrucción `scanf()` que obtiene el valor de `sec`. En particular, la primera vez que se realiza la prueba, el programa no ha tenido la oportunidad de obtener un valor por segundo, y la comparación se realizará con algún valor basura que se encuentre en esa ubicación de memoria; puede o no ser mayor que 0. Una solución, aunque poco elegante, es inicializar `sec` a, por ejemplo, 1 para que la prueba se pase la primera vez. Esto descubre un segundo problema. Cuando finalmente escribes 0 para detener el programa, `sec` no verifique hasta que termine el ciclo y se impriman los resultados de 0 segundos. Lo que realmente quiere es tener una instrucción `scanf()` justo antes de realizar la prueba. Puede lograr eso alterando la parte central del programa para leer de esta manera:

```
#include <stdio.h>
#define S_TO_M 60
int main(void)
{
        int sec, min, left;
        printf("This program convierte segundos a minutos y ");
        printf("segundos.\n");
        printf("Solo debe ingresar los segundos.\n");
        printf("Ingrese 0 para finalizar el programa.\n");
        scanf("%d", &sec);
        while(sec>0){
                min = sec/S_TO_M;
                left = sec % S_TO_M;
                printf("%d sec is %d min, %d sec.\n", sec, min, left);
                printf("Next input?\n");
                scanf("%d", &sec);
        }
        printf("See you!\n");

        return 0;
}
```

Listado 3: Programa `exercise4_3.c`.

La primera vez, se utiliza el `scanf()` fuera del bucle. A partir de entonces, se usa el `scanf()` al final del ciclo (y, por tanto, justo antes de que el ciclo comience de nuevo).

4. Escriba un programa que pregunte por un entero, y que imprima todos los enteros desde este número hasta 10 más de este. (Si fuera 5, sería desde 5 hasta 15). Asegúrese de separar cada valor de salida por un espacio o tabulador o una nueva línea.

Solución

```c
#include <stdio.h>
int main(void)
{
        int number, i=0;

        printf("Digite un número entero:\n");
        scanf("%d", &number);
        while(i<=10)
        {
                printf("%d\n", number + i);
                i++;
        }

        return 0;
}
```

Listado 4: Programa `exercise4_4.c`.

5. Escriba un programa que solicite un decimal e imprima su cubo.

Solución

```c
#include <stdio.h>
int main(void)
{
        float number;

        printf("Digite un número decimal:\n");
        scanf("%f", &number);
        printf("Su cubo es %.2f", number * number * number);

        return 0;
}
```

Listado 5: Programa `exercise4_5.c`.

6. Use un **`while`** loop para convertir el tiempo en minutos a el tiempo en horas y minutos.

Solución

```c
#include <stdio.h>
#define M_TO_H 60
int main(void)
{
        int min, hour, left;
        printf("This program convierte minutos a horas y ");
        printf("minutos.\n");
        printf("Solo debe ingresar los minutos.\n");
        printf("Ingrese 0 para finalizar el programa.\n");
        scanf("%d", &min);
        while(min>0){
                hour = min/M_TO_H;
                left = min % M_TO_H;
                printf("%d min is %d hour, %d min.\n", min, hour, left);
                printf("Next input?\n");
                scanf("%d", &min);
        }
        printf("See you!\n");

        return 0;
}
```

Listado 6: Programa `exercise4_6.c`.

7. ¿Qué imprimirá este programa?

```c
#include<stdio.h>
#define FORMAT "%s is a string\n"
main()
{
int num=0;
printf(FORMAT, FORMAT);
printf("%d\n", ++num);
printf("%d\n", num++);
printf("%d\n", num--);
printf("%d\n", num);
}
```

Solución

```c
#include <stdio.h>
#define FORMAT "%s is a string\n"
int main(void)
{
        int num = 0;

        printf(FORMAT,FORMAT);
        printf("%d\n", ++num);
        printf("%d\n", num++);
        printf("%d\n", num--);
        printf("%d\n", num);

        return 0;
}
```

Listado 7: Programa `exercise4_7.c`.

Luego de ejecutar el programa, se obtiene en consola el resultado:

```
pc@CTIC:~$ x: gcc exercise4_7.c && ./a.out
%s is a string
is a string
1
1
2
1
```

8. Cambie el programa `addemup.c` para calcular cuánto dinero ganaría en 20 días, si recibe 1$ el primer día, 2$ el segundo, 3$ el tercero y así.

Solución

```c
#include <stdio.h>
int main(void)
{
        int count, sum;

        count = 0;
        sum = 0;
        while (count++ < 20)
                sum = sum + count;

        printf("En %d días ganaría $%d.\n", 20, sum);

        return 0;
}
```

Listado 8: Programa `exercise4_8.c`.

9. Escriba un programa que convierta sus días en semanas y días.

Solución

```c
#include <stdio.h>
#define D_TO_W 7
int main(void)
{
        int day, week, left;
        printf("This program convierte días a semanas y ");
        printf("días.\n");
        printf("Solo debe ingresar los días.\n");
        printf("Ingrese 0 para finalizar el programa.\n");
        scanf("%d", &day);
        while(day>0){
                week = day/D_TO_W;
                left = day % D_TO_W;
                printf("%d días es %d semana(s), %d día(s).\n", day, week, left);
                printf("Next input?\n");
                scanf("%d", &day);
        }
        printf("See you!\n");

        return 0;
}
```

Listado 9: Programa `exercise4_9.c`.

10. Construya sentencia que hagan lo siguiente:

    a. Incremente la variable x por 10.

    b. Incremente la variable x por 1.

    c. Asigne dos veces la suma de a y b a c.

    d. Asigne a más dos veces b a c.

Solución

```c
#include <stdio.h>
int main(void)
{
        float x, a, b, c;
        x = x + 10;
        x++; ++x; x = x + 1;
        c = 2 * (a + b);
        c = a + 2* b;

        return 0;
}
```

Listado 10: Programa `exercise4_10.c`.

Código disponible en .