

Actividad | 3 | Código en Lenguaje C

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Felipe de Jesús Araux López

ALUMNO: Erik Antonio Reyes Gabriel

FECHA: 25 de julio de 2023

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo	6
Calculadora números primos	6
Calculadora números pares e impares	8
Calculadora al revés	12
Conclusión	14
Referencias	15

Introducción

a continuación, se mostrará lo que son 3 tipos de calculadoras, cada una con diferentes funciones, como puede observarse será todo referente a las actividades anteriores, fueron una secuencia de actividades para llegar hasta este punto, cada calculadora será programada en lenguaje C, compilado en programiz, un compilador online.

Como se ha visto en las actividades anteriores se ocuparon matemáticas lógicas para resolver cada problema que se presentaba para cada calculadora, ahora es momento de pasarlo a código para saber si estábamos en lo correcto hasta hacer correr el código y que nos pueda ser de funcionalidad.

En experiencia anterior a partir de aquí es donde se puede poner un poco complicado el asunto, solo que aquí ya aplicaremos lo aprendido en tutorías, en el material de apoyo y sobre todo en los sitios de internet o libros para poder tener un mayor amplio conocimiento para poder aplicar correctamente cada código y puedan correr los programas de manera funcional.

Descripción

En mi opinión lo que se nos pide realizar en esta actividad podría ser difícil para los compañeros que aún no tienen un conocimiento como tal del tema, siento que se les podrá tornar un poco complicado el investigar por otros medios el incluso como empezar un algoritmo, pero también considero que este tipo de actividades sencillas los ara sentir el amor por la programación. La actividad como tal no es tan complicada, si lo vemos en un punto de vista de las actividades anteriores, es fácil detectar un numero par, un numero primo o poner un numero al revés, pero pasando programar en lenguaje C, o cualquier otro tipo de lenguaje ya es más diferente ya que estas a prueba y error ya que si ponemos mal un código el programa no funciona, hoy en día muchos compiladores te indican cual línea tienes error, ese es una gran ventaja.

Dentro de esta actividad se mostrarán un ciclo, sentencias de si o no, todo esto visto ya en tutoría y el material de apoyo.

Justificación

Empezar a programar en C este tipo de programas nos ayudara en un futuro para ir entrando en confianza en el mundo de programación ya que es mucho el programar y nos facilitara a utilizar menos códigos en un algoritmo, ya que entre menos código más fácil nos será encontrar un error en nuestro código fuente cuando este no corra de manera correcta.

Considero que también saber cómo es la estructura del lenguaje utilizado, como se inicia, como declarar variable, las sentencias, en fin todo lo que conlleva el utilizar un lenguaje de programación, ya que podemos tener el conocimiento teórico, pero al momento de ponerlo en práctica si no llegas a tener paciencia y no saber dónde poner cada cosa en su lugar traerá muchos dolores de cabeza, ya que por cosas tan simples como punto y coma, corchetes mal colocados puede afectar el código y no correrlo de la forma que debería y solo manda error.

Igual el uso del inglés podría considerarse necesario en programación ya que la gran parte de la estructura del código es en inglés, y el saber por lo menos las palabras reservadas o las sentencias a utilizar nos será de gran valor.

Desarrollo

Se mostrarán 3 diferentes códigos en C de las 3 diferentes calculadoras

Calculadora de números primos.

Se muestra un código en lenguaje **c**, donde nos pide ingresar un número y que el programa nos arroje la leyenda si es número primo o no, donde se ocupó la lógica matemática de entrada donde el numero ingresado se va a dividir la cantidad de veces hasta llegar al número ingresado, por ejemplo si es el número 4, se divide entre 1, entre 2, entre 3, entre 4, y si su residuo es 0 en 2 ocasiones es un numero primo si no, no lo es, esto usando 2 contenedores, para hacer todo el proceso, a continuación se muestra el código completo en **c** y las capturas donde si funciona el programa en el compilador programiz.

```
#include<stdio.h>
int main ()
{
    int a, i, c;
    i = 1;
    c= 0;
    printf("introduce 1 numero entero:\n");
    scanf ("%i", &a);
    while (i <= a)
    {
        if (a % i==0)
        {
            c++;
        }
        i++;
    }
    if (c == 2)
    {
        printf ("el numero es primo");
    }
    else
    {
        printf ("el numero no es primo");
    }
    return 0;
}
```

Programación en C Online C Compiler

programiz.com/c-programming/online-compiler/

main.c

```
1 // Online C compiler to run C program online
2 #include<stdio.h>
3 int main ()
4 {
5     int a, i, c;
6     i = 1;
7     c = 0;
8     printf("introduce 1 numero entero:\n");
9     scanf ("%i", &a);
10    while (1 <= a)
11    {
12        if (a % i==0)
13        {
14            c++;
15        }
16        i++;
17    }
18    if (c == 2)
19    {
20        printf ("el numero es primo");
21    }
22 }
```

Output

Clear

Run

28°C Mayorm. nublado

Buscar

09:44 a. m. 30/07/2023

Programiz C Online Compiler

main.c

```
1 // Online C compiler to run C program online
2 #include<stdio.h>
3 int main ()
4 {
5     int a, i, c;
6     i = 1;
7     c = 0;
8     printf("introduce 1 numero entero:\n");
9     scanf ("%i", &a);
10    while (1 <= a)
11    {
12        if (a % i==0)
13        {
14            c++;
15        }
16        i++;
17    }
18    if (c == 2)
19    {
20        printf ("el numero es primo");
21    }
22 }
```

Output

Clear

Run

/tmp/voY8bDW5TT.o
introduce 1 numero entero:
1
el numero no es primo

Programiz C Online Compiler

main.c

```
1 // Online C compiler to run C program online
2 #include<stdio.h>
3 int main ()
4 {
5     int a, i, c;
6     i = 1;
7     c = 0;
8     printf("introduce 1 numero entero:\n");
9     scanf ("%i", &a);
10    while (1 <= a)
11    {
12        if (a % i==0)
13        {
14            c++;
15        }
16        i++;
17    }
18    if (c == 2)
19    {
20        printf ("el numero es primo");
21    }
22 }
```

Output

Clear

Run

/tmp/voY8bDW5TT.o
introduce 1 numero entero:
7
el numero es primo

Calculadora de pares impares.

Se mostrara el código en C de un programa que pida al usuario ingresar 10 números desde el teclado y el programa detecte cuales son números pares y cuáles serán numeros impares, usando como lógica la división de los números ingresados y si su residuo es igual a 0 se manda a imprimir en pantalla que es un numero par, si no por consiguiente es un numero impar, si una sentencia se cumple sigue con el siguiente número así hasta tener el total de los 10 números, a continuación se mostrara el código en C y las capturas de pantallas donde el programa funciona correctamente.

```
#include <stdio.h>
int main() {
    int a, b, c, d, e, f, g, h, i, j;
    printf ("ingresa 10 numeros:");
    scanf ("%i %i %i %i %i %i %i %i %i %i", &a, &b, &c, &d, &e, &f, &g, &h, &i, &j);
    if (a % 2 == 0){
        printf ("es par %i\n", a);
    }
    else {
        printf("es impar %i\n", a);
    }
    if (b % 2 == 0)
    {
        printf ("es par %i\n", b);
    }
    else {
        printf("es impar %i \n", b);
    }
    if (c % 2 == 0){
        printf ("es par %i\n", c);
    }
    else {
        printf("es impar %i\n", c);
    }
    if (d % 2 == 0)
    {
        printf ("es par %i\n", d);
    }
    else {
        printf("es impar %i\n", d);
    }
    if (e % 2 == 0){
        printf ("es par %i\n", e);
    }
    else {
        printf("es impar %i\n", e);
    }
}
```



```

}
if (f % 2 ==0)
{
printf ("es par %i\n", f);
}
else {
printf("es impar %i\n", f);
}
if (g % 2 ==0){
printf ("es par %i\n", g);
}
else {
printf("es impar %i\n", g);
}
if (h % 2 ==0)
{
printf ("es par %i\n", h);
}
else {
printf("es impar %i\n", h);
}
if (i % 2 ==0){
printf ("es par %i\n", i);
}
else {
printf("es impar %i\n", i);
}
if (j % 2 ==0)
{
printf ("es par %i\n", j);
}
else {
printf("es impar %i\n", j);
}
return 0;
}

```

Programiz C Online Compiler

C Certification >

main.c

Run

Output

Clear

```
2 #include <stdio.h>
3
4 int main() {
5     int a, b, c, d, e, f, g, h, i, j;
6     printf ("ingresa 10 numeros:");
7     scanf ("%i %i %i %i %i %i %i %i %i %i" ,&a, &b,&c, &d,&e, &f,&g,
8         &h,&i, &j);
9     if (a % 2 ==0){
10         printf ("es par %i\n", a);
11     }
12     else {
13         printf("es impar %i\n", a);
14     }
15     if (b % 2 ==0)
16     {
17         printf ("es par %i\n",b);
18     }
19     else {
20         printf("es impar %i\n", b);
21     }
```

Programiz C Online Compiler

C Certification >

main.c

Run

Output

Clear

```
20 }
21 if (c % 2 ==0){
22     printf ("es par %i\n", c);
23 }
24 else {
25     printf("es impar %i\n", c);
26 }
27 if (d % 2 ==0)
28 {
29     printf ("es par %i\n", d);
30 }
31 else {
32     printf("es impar %i\n", d);
33 }
34 if (e % 2 ==0){
35     printf ("es par %i\n", e);
36 }
37 else {
38     printf("es impar %i\n", e);
39 }
40 if (f % 2 ==0)
```

Programiz C Online Compiler

C Certification >

main.c

Run

Output

Clear

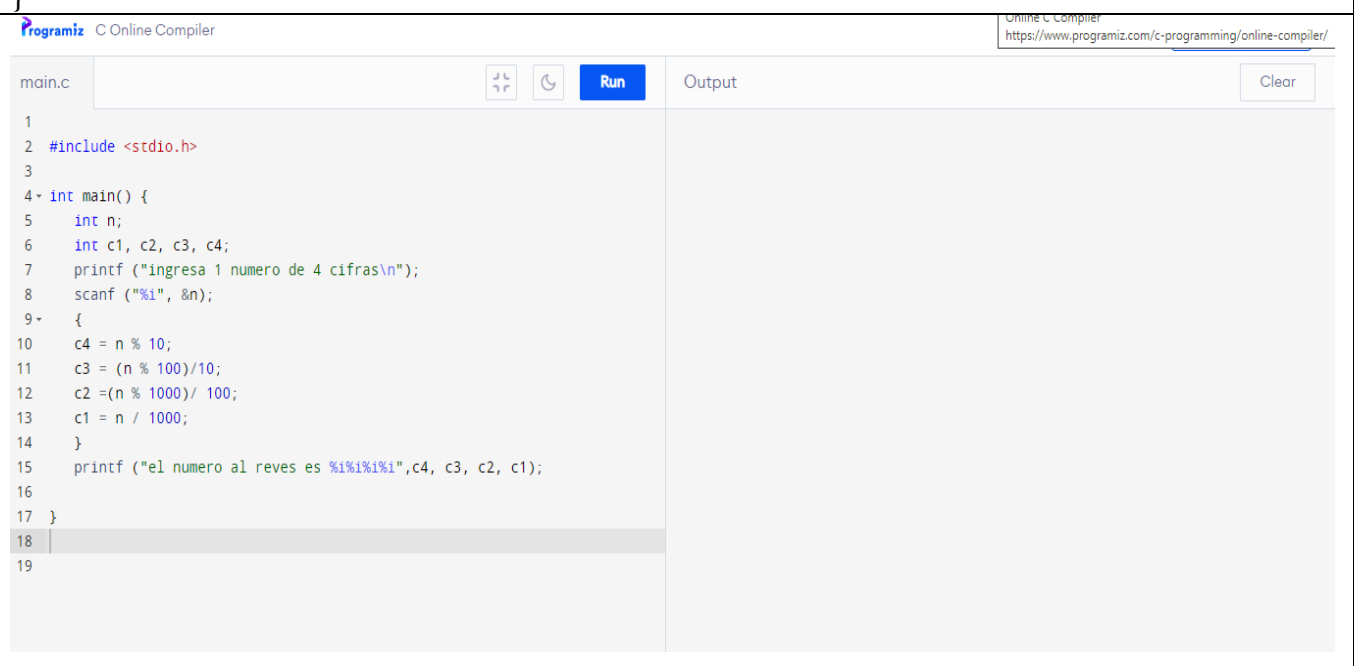
```
43 }
44 else {
45     printf("es impar %i\n", f);
46 }
47 if (g % 2 ==0){
48     printf ("es par %i\n", g);
49 }
50 else {
51     printf("es impar %i\n", g);
52 }
53 if (h % 2 ==0)
54 {
55     printf ("es par %i\n", h);
56 }
57 else {
58     printf("es impar %i\n", h);
59 }
60 if (i % 2 ==0){
61     printf ("es par %i\n", i);
62 }
63 else {
```


Calculadora de numero al revés.

Por ultimo se muestra el ultimo programa que pide al usuario ingresar un numero de 4 dígitos donde al momento de imprimir lo muestre el programa al revés, de derecha a izquierda, esto solo ocupamos que el programa lea el numero y lo divida en dígitos, los 4 y para sacar cada digito se aplica una división por digito, para que nos de solo el numero que necesitamos, una vez realizado mandamos imprimir el programa desde el digito 4 hasta el digito uno como ultimo.

A continuación, se muestra el código en C y las capturas donde el programa funciona correctamente.

```
#include <stdio.h>
int main() {
    int n;
    int c1, c2, c3, c4;
    printf ("ingresa 1 numero de 4 cifras\n");
    scanf ("%i", &n);
    {
        c4 = n % 10;
        c3 = (n % 100)/10;
        c2 =(n % 1000)/ 100;
        c1 = n / 1000;
    }
    printf ("el numero al revés es %i%i%i%i",c4, c3, c2, c1);
}
```



Programiz C Online Compiler

C Certification >

main.c

Run

```
1
2 #include <stdio.h>
3
4 int main() {
5     int n;
6     int c1, c2, c3, c4;
7     printf ("ingresa 1 numero de 4 cifras\n");
8     scanf ("%i", &n);
9     {
10        c4 = n % 10;
11        c3 = (n % 100)/10;
12        c2 =(n % 1000)/ 100;
13        c1 = n / 1000;
14    }
15    printf ("el numero al revés es %i%i%i%i",c4, c3, c2, c1);
16
17 }
18
19
```

Output

Clear

```
/tmp/PxVpdzbewv.o
ingresa 1 numero de 4 cifras
8765
```

Programiz C Online Compiler

C Certification >

main.c

Run

```
1
2 #include <stdio.h>
3
4 int main() {
5     int n;
6     int c1, c2, c3, c4;
7     printf ("ingresa 1 numero de 4 cifras\n");
8     scanf ("%i", &n);
9     {
10        c4 = n % 10;
11        c3 = (n % 100)/10;
12        c2 =(n % 1000)/ 100;
13        c1 = n / 1000;
14    }
15    printf ("el numero al revés es %i%i%i%i",c4, c3, c2, c1);
16
17 }
18
19
```

Output

Clear

```
/tmp/PxVpdzbewv.o
ingresa 1 numero de 4 cifras
8765
el numero al revés es 5678
```

Conclusión

Como conclusión podría decirse que el uso del lenguaje **C** es un poco complicado para ciertas cosas, pero igual agarrando el ritmo para empezar a programar se empieza a tornar cada vez más fácil, en la experiencia que he tenido anteriormente programando, considero que el compilador programiz es fácil de utilizar, aunque hay otros que podrían considerarse más fáciles, pero para empezar es un buen programa, ocupar lenguaje **C** como inicio nos abrirá un camino a lo que nos espera en un futuro en la carrera, ya que si aún no se sabe si se ocupara otro lenguaje de programación, esto nos encamina a las dificultades y ventajas que tendremos con los diferentes tipos de lenguajes, ya que podría considerar que es de prueba y error, ya que si no sabemos dónde está un error en el código se nos complicara todo lo demás, incluso sé que ya hay programas que te pueden hacer un código pero eso no es programación, aquí estamos para enfrentar estos problemas y ocupar la lógica para afrontarlos.

Referencias.

Se usaron como referencias las siguientes paginas y videos de apoyo.

*Parzibyte. (2019). Saber si número es par o impar en c. *Parzibyte's blog*.

<https://parzibyte.me/blog/2019/10/09/numero-par-impar-c/#:~:text=Primero%20solicitamos%20un%20n%C3%BAmero%20con,se%20dice%20que%20es%20impar.>

*DiloenTutos Pc. (2017, 17 septiembre). *PSEINT - Determine si un numero es par o impar* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=WaDbKaZgeTA>

*Rivera, C. (2022, 14 marzo). Números primos y compuestos: qué son + ejemplos Smartick. *Smartick*. <https://www.smartick.es/blog/matematicas/numeros-enteros/numeros-primos-y-numeros>

**Forum - Grupo de C++ - Azul School*. (s. f.). Azul School. <https://www.azulschool.net/todos-los-grupos/grupo-de-c/forum/topic/como-saber-si-un-numero-es-primo-utilizando>.

*Programa Resuelto. (2022b, mayo 4). *#17 [C] – Leer un número de cuatro cifras e imprimirlo al revés* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=aIAIGNZJ2iE>

*DiloenTutos Pc. (2017, 29 septiembre). *Pseint - determinar si un número es primo* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=1VShvozLXvw>