	Estructuras de Repetición	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación
salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 10

Integrante(s): Lorena Basurto Amezcu

*No. de Equipo de
cómputo empleado:* Mónaco 21

No. de Lista o Brigada: 2858

Semestre: 2020-1

Fecha de entrega: Octubre 28, 2019.

Observaciones: Muy bien

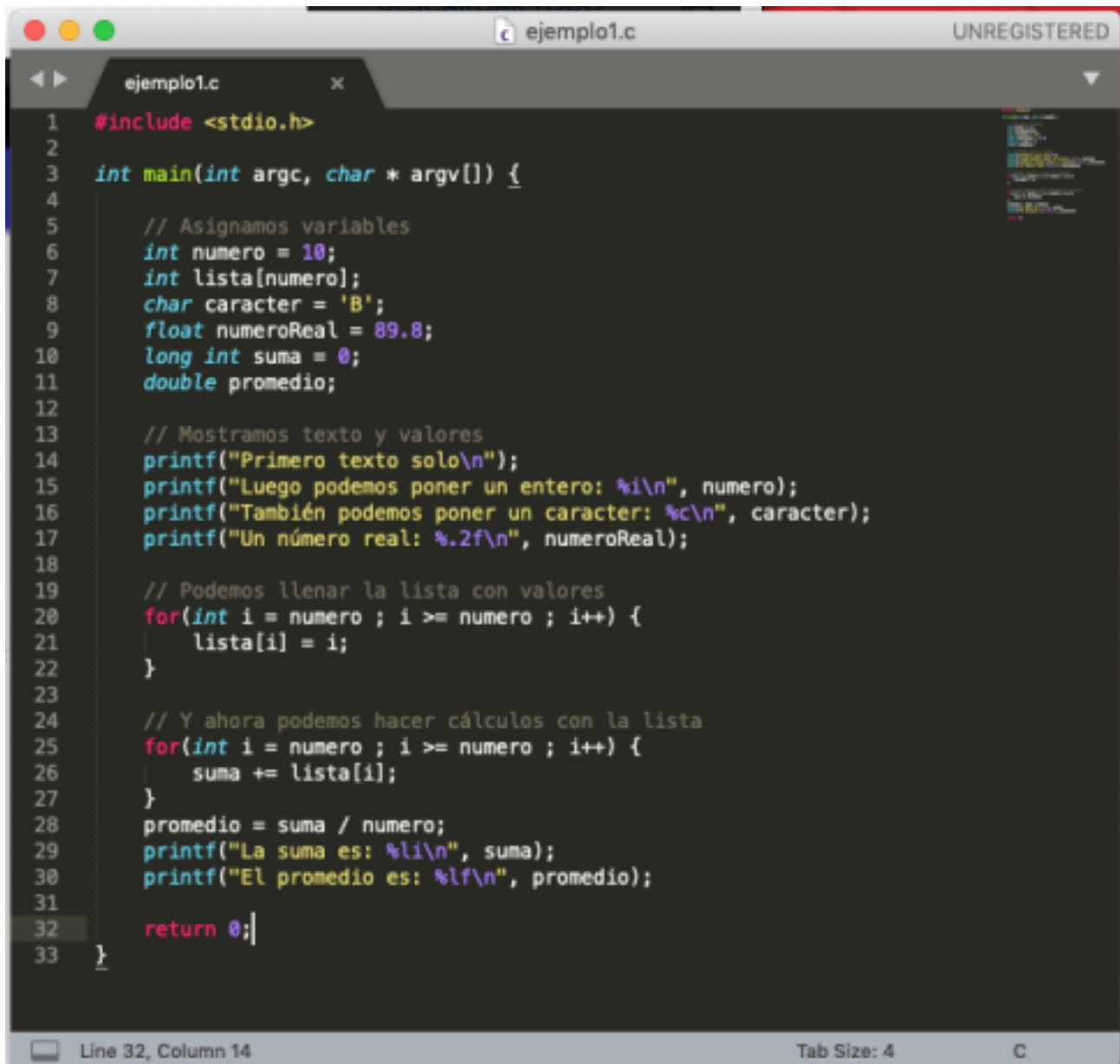
CALIFICACIÓN: **10**

Objetivo:

Aprender las técnicas básicas de depuración de programas en C para revisar de manera precisa el flujo de ejecución de un programa y el valor de las variables; en su caso, corregir posibles errores.

Desarrollo:

- Ejemplo 1



```
1  #include <stdio.h>
2
3  int main(int argc, char * argv[]) {
4
5      // Asignamos variables
6      int numero = 10;
7      int lista[numero];
8      char caracter = 'B';
9      float numeroReal = 89.8;
10     long int suma = 0;
11     double promedio;
12
13     // Mostramos texto y valores
14     printf("Primero texto solo\n");
15     printf("Luego podemos poner un entero: %i\n", numero);
16     printf("También podemos poner un caracter: %c\n", caracter);
17     printf("Un número real: %.2f\n", numeroReal);
18
19     // Podemos llenar la lista con valores
20     for(int i = numero ; i >= numero ; i++) {
21         lista[i] = i;
22     }
23
24     // Y ahora podemos hacer cálculos con la lista
25     for(int i = numero ; i >= numero ; i++) {
26         suma += lista[i];
27     }
28     promedio = suma / numero;
29     printf("La suma es: %li\n", suma);
30     printf("El promedio es: %lf\n", promedio);
31
32     return 0;
33 }
```

Line 32, Column 14 Tab Size: 4 C

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 91x39
Last login: Mon Oct 14 09:52:55 on ttys000
[Surinam46:~ fp03alu05$ servidor
mkdir: /Users/fp03alu05/Desktop/fp03alu05: File exists
mount_smbfs: mount error: /Users/fp03alu05/Desktop/fp03alu05: File exists
[Surinam46:~ fp03alu05$ ssh fp03alu05@192.168.2.200
The authenticity of host '192.168.2.200 (192.168.2.200)' can't be established.
RSA key fingerprint is SHA256:jTgFsbvP7IaIpwchV27DaUa9i2pvAVVZwZzbIneOF8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
fp03alu05@192.168.2.200's password:
Last login: Mon Oct 7 13:29:32 2019 from inglaterra.red

Samba

-bash: aviso: setlocale: LC_CTYPE: no se puede cambiar el local (UTF-8)
[[fp03alu05@samba ~]$ ls
Escritorio hola hola.c
[[fp03alu05@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
gcc: error: ejemplo1.c: No existe el fichero o el directorio
gcc: error fatal: no hay ficheros de entrada
compilaci?n terminada.
[[fp03alu05@samba ~]$ ls
Escritorio hola hola.c
[[fp03alu05@samba ~]$ gcc -std=c99 -g ejemplo1.c -o ejemplo1
[[fp03alu05@samba ~]$ ./ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
Tambi?n podemos poner un caracter: B
Un n?mero real: 89.89
Violaci?n de segmento ('core' generado)
[[fp03alu05@samba ~]$
```

```

[fp03alu05@samba ~]$ gdb ./ejemplo1
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu05/ejemplo1...done.
(gdb) run
Starting program: /users/fp03/fp03alu05/ejemplo1
Primero texto solo
Luego podemos poner un entero: 10
También podemos poner un caracter: B
Un número real: 89.80

Program received signal SIGSEGV, Segmentation fault.
0x000000000040060c in main (argc=19, argv=0x1100000010) at ejemplo1.c:21
21             lista[i] = i;
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) list
16             printf("También podemos poner un caracter: %c\n", caracter);
17             printf("Un número real: %.2f\n", numeroReal);
18
19             // Podemos llenar la lista con valores
20             for(int i = numero ; i >= numero ; i++) {
21                 lista[i] = i;
22             }
23
24             // Y ahora podemos hacer cálculos con la lista
25             for(int i = numero ; i >= numero ; i++) {
(gdb) █

```

- **Actividad 1**

Utilizar GDB para encontrar la utilidad del programa y describir su funcionalidad.

GDB nos permite recorrer línea por línea nuestro programa y saber qué está haciendo en cada paso, incluso al recibir valores del usuario y utilizarlos en una iteración como sucede en el caso del código de la actividad 1.

El código de la actividad 1 inicializa la variable *AS* en 0 para hacer sumas de $N/2$ (o $N/2 + 1/2$ si N es impar) números impares, empezando por el 1, utilizando un contador *CONT*. Este código no tiene errores y se puede observar su funcionamiento utilizando el comando *gdb* y *ctrl+x+a* en la terminal.

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x26

actividad1.c
1 #include <stdio.h>
2
3 void main()
4 {
5     int N, CONT, AS;
6     AS=0;
7     CONT=1;
8     printf("Ingresa un número: ");
9     scanf("%i",&N);
> 10 while(CONT<=N)
11 {
12     AS=(AS+CONT);
13     CONT=(CONT+2);
14 }

Ingresa un número: 6
Starting program: /users/fp03/fp03alu05/actividad1
(gdb) start
Temporary breakpoint 1 at 0x4005f4: file actividad1.c, line 6.
Starting program: /users/fp03/fp03alu05/actividad1

Temporary breakpoint 1, main () at actividad1.c:6
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb)
n
```

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x26

actividad1.c
[ No Source Available ]
18
19
20

El resultado es: 9 In: __libc_start_main Line: ?? PC: 0x3c6ba21735
0x0000003c6ba21735 in __libc_start_main () from /lib64/libc.so.6
(gdb)
```

- **Actividad 2**
Utilizar GDB para corregir el programa.

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x52
—actividad2.c—
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      int K, AP, N;
7      double X, AS;
8  B+> printf("Ingrese cuántos términos calcular de la serie: X^
9      printf("\nN=");
10     scanf("%i",N);
11     printf("X=");
12     scanf("%lf",X);
13     K=0;
14     AP=1;
15     AS=0;
16     while(K<=N)
17     {
18         AS=AS+pow(X,K)/AP;
19         K=K+1;
20         AP=AP*K;
21     }
22     printf("Resultado=%le",AS);
23 }
24
25
26
child process 17839 In: main Line: 8 PC: 0x400664
(gdb) start
The program being debugged has been started already.
Start it from the beginning? (y or n)

Temporary breakpoint 1 at 0x400664: file actividad2.c, line 8.
Starting program: /users/fp03/fp03alu05/actividad2

Temporary breakpoint 1, main () at actividad2.c:8
(gdb) □
```

```
actividad2.c
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      int K, AP, N;
7      double X, AS;
8      printf("Ingrese cuantos términos calcular de la serie: X^
> 9      printf("\nN=");
10     scanf("%i",N);
11     printf("X=");
12     scanf("%lf",X);
13     K=0;
14     AP=1;
15     AS=0;
16     while(K<=N)
17     {
18         AS=AS+pow(X,K)/AP;
19         K=K+1;
20         AP=AP*K;
21     }
22     printf("Resultado=%le",AS);
23 }
24
25
26
```

child process 17839 In: main Line: 9 PC: 0x400673

(gdb) start

The program being debugged has been started already.

Start it from the beginning? (y or n)

Temporary breakpoint 1 at 0x400664: file actividad2.c, line 8.

Starting program: /users/fp03/fp03alu05/actividad2

Temporary breakpoint 1, main () at actividad2.c:8

(gdb) █

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x52
actividad2.c
1  #include <stdio.h>
2  #include <math.h>
3
4  void main()
5  {
6      int K, AP, N;
7      double X, AS;
8      printf("Ingrese cuántos términos calcular de la serie: X^
9      printf("\nN=");
> 10     scanf("%i",N);
11     printf("X=");
12     scanf("%lf",X);
13     K=0;
14     AP=1;
15     AS=0;
16     while(K<=N)
17     {
18         AS=AS+pow(X,K)/AP;
19         K=K+1;
20         AP=AP*K;
21     }
22     printf("Resultado=%le",AS);
23 }
24
25
26
Ingrese cuántos términos calcular de la serie: X^K/K!
child process 17839 In: main                               Line: 10    PC: 0x400682
(gdb) start
The program being debugged has been started already.
Start it from the beginning? (y or n)

Temporary breakpoint 1 at 0x400664: file actividad2.c, line 8.
Starting program: /users/fp03/fp03alu05/actividad2

Temporary breakpoint 1, main () at actividad2.c:8
(gdb) █
```

```
N=87ese cuántos términos calcular de la serie: X^K/K!
child process 18010 In: _IO_vfscanf_internal               Line: ??    PC: 0x3c6ba5940d
(gdb) start
Temporary breakpoint 1 at 0x400664: file actividad2.c, line 8.
Starting program: /users/fp03/fp03alu05/actividad2

Temporary breakpoint 1, main () at actividad2.c:8
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb)
Program received signal SIGSEGV, Segmentation fault.
0x0000003c6ba5940d in _IO_vfscanf_internal () from /lib64/libc.so.6
(gdb) █
```



```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x29
-actividad2.c
5      {
6          int K, AP, N;
7          double X, AS;
8          printf("Ingrese cuántos términos calcular de la serie: X^
9          printf("\nN=");
10         scanf("%i",&N);
11         printf("X=");
12         scanf("%lf",&X);
24 13         K=0;
14         AP=1;
15         AS=0;
16         while(K<=N)
17         {
18             AS=AS+pow(X,K)/AP;
19             K=K+1;
Resultado=1.850000e+01
child No process in: Line: ?? PC: ??

0x0000003c6ba21735 in __libc_start_main () from /lib64/libc.so.6
(gdb) n
n
Single stepping until exit from function __libc_start_main,
which has no line numb[Inferior 1 (process 18894) exited with code 026]
(gdb) █
```

- **Actividad 3**

Utilizar GDB para corregir el programa.

```
[fp03alu05@samba ~]$ gcc -g -std=c99 actividad3.c -o actividad3 -lm
[fp03alu05@samba ~]$ gdb ./actividad3
GNU gdb (GDB) Fedora (7.4.50.20120120-42.fc17)
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /users/fp03/fp03alu05/actividad3...done.
(gdb) run
Starting program: /users/fp03/fp03alu05/actividad3
Ingrese un número:
5
El factorial de -1 es 0.
[Inferior 1 (process 19218) exited normally]
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) █
```

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x50

actividad3.c
5      int numero;
6
7      printf("Ingrese un número:\n");
8      scanf("%i",&numero);
9
10     long int resultado = 1;
11     while(numero>=0){
12         numero--;
13         resultado *= numero;
14     }
15
16     printf("El factorial de %i es %li.\n", numero, resultado);
17
> 18     return 0;
19 }
20
21
22
23
24
25
26
27
28
29

El factorial de -1 es 0.
Child process 19228 In: main                                Line: 18    PC: 0x4006a7
```

```
fp03alu05 — fp03alu05@samba:~ — ssh fp03alu05@192.168.2.200 — 80x34

actividad3.c
12
13     long int resultado = 1;
14     while(numero>0){
15         resultado *= numero;
16         numero--;
17     }
18
19     printf("El factorial de %i es %li.\n", copiaNumero, resultado);
20
> 21     return 0;
22 }
23
24
25
26
27
28

El factorial de 6 es 720.
```

```
Ingrese un número:
5
El factorial de 5 es 120.
(Inferior 1 (process 19650) exited normally)
Missing separate debuginfos, use: debuginfo-install glibc-2.15-37.fc17.x86_64
(gdb) █
```

Conclusiones:

GDB es una herramienta muy útil para analizar a detalle el funcionamiento de un programa, lo que facilita su comprensión, así como el encontrar posibles fallas en el mismo.