TALLER00

Integrantes: Miguel Angel Rico Llanos, Edward Quintero, Mateo Maldonado

5.1 Compilación

1. ¿Qué contiene cada uno de los archivos?

- Exercise1.cxx: Inicialmente se puede observar que en el código se incluye la interfaz llamada “rectangle.h”, posteriormente en el main del código se crea un objeto de tipo Rectangle (Es una estructura) llamado rect1 y se declaran unas variables a,b. Procede a pedir por consola los atributos de rectángulo es decir; Posición en X, posición en Y, el ancho y el alto del rectángulo.

Para finalizar con el main, hace llamado a las funciones “perimeterRect” para hallar el perímetro, “AreaRect” para hallar el área y por último, “distOriginRect” para encontrar la distancia del rectángulo al origen de las coordenadas.

-Rectangle.h: Inicialmente se define la interfaz usando “#define RECTANGLE_H” posteriormente, se crea la estructura “Rectangle” que contiene los siguientes atributos: “posX,posY,width,height”. Por último, contiene las firmas de cada una de las funciones llamadas en el main (Exercise1.cxx).

-Rectangle.cxx: Primero incluye la librería que llama a la interfaz “Rectangle.h”, continuando con el archivo se encuentran las definiciones de cada función:

“perimeterRect” encargada de calcular el perímetro del rectángulo sumando el ancho con el alto y multiplicándolo por 2, para posteriormente retornar el perímetro.

“areaRect” encargada de calcular el área del rectángulo sumando el ancho por el alto para retornar el área al final.

“distOriginRect” esta función se encarga de para encontrar la distancia del rectángulo al origen de las coordenadas haciendo el siguiente procedimiento, la raíz cuadrada de posición de X al cuadrado mas la posición de Y al cuadrado para al final retornar la distancia.

2. ¿Cuál es el comando completo a escribir en la consola que permite compilar y encadenar estos archivos sin errores?

```
~/Taller00$ g++ -std=c++11 -g -o exercise1 exercise1.cxx rectangle.cxx
```

Imagen 1: Muestra el código utilizado para compilar y encadenar los archivos por consola.

3. ¿Cuál es el comando que permite ejecutar en la consola el programa anteriormente compilado?

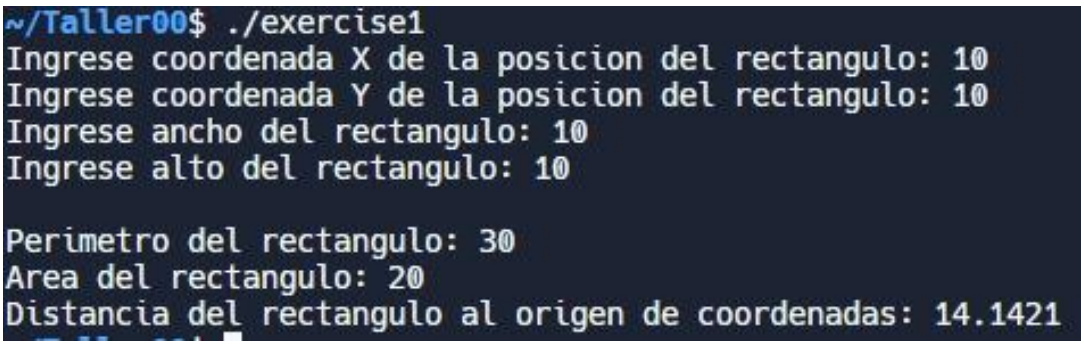

Imagen 2: Hace referencia al comando utilizado para ejecutar el programa y su funcionamiento.

5.2 Plan de Pruebas

1. Defina un plan de pruebas (en el formato ya explicado) para la operación "perímetro del rectángulo" en el programa. Incluya valores nulos y/o negativos en los casos de prueba.

Plan de Pruebas: Función hallar perímetro			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1: Números Positivos	a.20 b. 10	60	50
2: Un número 0	a.15 b.0	30	30
3: Números Negativos	a.22 b. -30	-16	14

2. Defina un plan de pruebas (en el formato ya explicado) para la operación "área del rectángulo" en el programa. Incluya valores nulos y/o negativos en los casos de prueba.

Plan de Pruebas: Función hallar area			
Descripción de caso	Valores de entrada	Resultado esperado	Resultado obtenido
1: Números Positivos	a.20 b. 10	200	30
2: Un número 0	a.15 b. 0	0	15
3: Números Negativos	a.22 b.-30	-660	-8

3. ¿Las pruebas fueron exitosas? En caso negativo, indique las modificaciones que requiere el código fuente (indicando archivo, número de línea y contenido) para que los resultados sean aceptables.

Las pruebas al ejecutar varias veces el programa arrojaron un resultado negativo debido a que los resultados obtenidos fueron diferentes a los esperados por el programa y sus funciones.

- El primer cambio que se requiere hacer para arreglar el archivo y obtener los resultados esperados es en el archivo “rectangle.cxx” en la línea 7, es decir en la función de hallar el perímetro “perimeterRect”, ya que a la hora de calcular el perímetro carece de los paréntesis que le dan prioridad a la suma y por ende se realiza primero la multiplicación con el ancho y luego suma el alto.
- Otro cambio identificado es en el archivo “rectangle.cxx” en la línea 14, en la función de hallar el área “areaRect”, el error identificado es a la hora de calcular el área ya que la función y el procedimiento planteados son erróneos ya que la función correcta seria multiplicar el ancho por el alto y no sumarlos como se encuentra planteado en el código.

```
~/Taller00$ ./exercise1
Ingrese coordenada X de la posicion del rectangulo: 15
Ingrese coordenada Y de la posicion del rectangulo: 30
Ingrese ancho del rectangulo: 20
Ingrese alto del rectangulo: 10

Perimetro del rectangulo: 50
Area del rectangulo: 30
Distancia del rectangulo al origen de coordenadas: 33.541
~/Taller00$ ./exercise1
Ingrese coordenada X de la posicion del rectangulo: 10
Ingrese coordenada Y de la posicion del rectangulo: 10
Ingrese ancho del rectangulo: 15
Ingrese alto del rectangulo: 0

Perimetro del rectangulo: 30
Area del rectangulo: 15
Distancia del rectangulo al origen de coordenadas: 14.1421
~/Taller00$ ./exercise1
Ingrese coordenada X de la posicion del rectangulo: 10
Ingrese coordenada Y de la posicion del rectangulo: 10
Ingrese ancho del rectangulo: 22
Ingrese alto del rectangulo: -30

Perimetro del rectangulo: 14
Area del rectangulo: -8
Distancia del rectangulo al origen de coordenadas: 14.1421
~/Taller00$
```

Imagen 3: Muestra los datos ingresados para hacer el plan de prueba.

5.3 Depuración

1. ¿Qué contiene el archivo?

Inicialmente se define una plantilla para la creación de una clase” T” (Clase General) la cual se encarga de la creación y destrucción de nodos e inserción y el establecimiento de valores en dichos nodos. Además, se incluye una clase para definir listas que incluyen los nodos y funciones para borrar los nodos de las listas.

2. ¿Cuál es el comando completo a escribir en la consola que permite compilar y encadenar el archivo sin errores?

```
~/Taller00$ g++ -std=c++11 -g -o exercise2 exercise2.cpp
```

Imagen 4: : Muestra el código utilizado para compilar y encadenar el archivo “exercise2”.cxx por consola

3. ¿Cuál es el comando que permite ejecutar en la consola el programa anteriormente compilado?

```
~/Taller00$ ./exercise2
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Creating Node, 5 are in existence right now
Destroying Node, 4 are in existence right now
4
3
2
1

Segmentation fault (core dumped)
```

Imagen 5: Comando para ejecutar el programa.

4. ¿El programa corre sin errores? Si es así, hasta aquí llega el ejercicio. En caso contrario, es necesario depurar el código en busca de la fuente del error. Para eso, responda en su reporte las siguientes preguntas:

5. ¿Cuál es el comando completo a escribir en la consola que permite compilar y encadenar el archivo para su uso dentro del depurador?

```
~/Taller00$ gdb exercise2
GNU gdb (GDB) 12.1
Copyright (C) 2022 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-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from exercise2...
(gdb) rub
Undefined command: "rub".  Try "help".
(gdb) run
Starting program: /home/runner/Taller00/exercise2
warning: Error disabling address space randomization: Operation not permitted
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/nix/store/4nlgxhb09sdr51nc9hdm8az5b08vzkgx-glibc-2.35-163/li
b/libthread_db.so.1".
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Creating Node, 5 are in existence right now
Destroying Node, 4 are in existence right now
4
3
2
1
```

6. Al depurar el programa dentro de gdb, ¿cuál es el error que se genera y en qué parte del código? ¿De qué forma puede corregirse el problema para que el programa se ejecute satisfactoriamente (sin errores)?

```
Program received signal SIGSEGV, Segmentation fault.
LinkedList<int>::remove (this=0x16aeeb0, item_to_remove=@0x7ffc283ab504: 1) at exercise2.cpp:8
8
88          marker = marker->next();
(gdb)
```