SECURE CODING LAB - II

Name: Mohammed Tousif
Roll : CB.EN.P2CYS22008

1. Debugging a program with logic error.

Taking a file broken.cpp to demonstrate gdb debug.

Broken.cpp :-

```
#include <iostream>
#include <cmath>

using namespace std;

int ComputeFactorial(int number) {
  int fact = 1;

for (int j = 1; j <= number; j++) {
        fact = fact * j;
}

return fact;
}

double ComputeSeriesValue(double x, int n) {
double seriesValue = 0.0;
 double xpow = 1;

 for (int k = 0; k <= n; k++) {
seriesValue += xpow / ComputeFactorial(k);
 xpow = xpow * x;
 }

return seriesValue;
}

int main() {
```

```
        cout << "This program is used to compute the value of the following series :
" << endl;

        cout << "(x^0)/0! + (x^1)/1! + (x^2)/2! + (x^3)/3! + (x^4)/4! + ..... +
(x^n)/n! " << endl;

        cout << "Please enter the value of x : " ;

         double x;
        cin >> x;

        int n;
        cout << endl << "Please enter an integer value for n : " ;
        cin >> n;
        cout << endl;

        double seriesValue = ComputeSeriesValue(x, n);
        cout << "The value of the series for the values entered is "
                << seriesValue << endl;

         return 0;
        }
```

Process & Output :-

```
tousif@TousifVM:~/Desktop$ g++ -g broken.cpp -o broken
tousif@TousifVM:~/Desktop$ ./broken
This program is used to compute the value of the following series :
(x^0)/0! + (x^1)/1! + (x^2)/2! + (x^3)/3! + (x^4)/4! + ........ + (x^n)/n!
Please enter the value of x : 2

Please enter an integer value for n : 3

The value of the series for the values entered is inf
tousif@TousifVM:~/Desktop$ g++ -g broken.cpp -o broken
tousif@TousifVM:~/Desktop$ gdb broken
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
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-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 broken...
(gdb) b 43
Breakpoint 1 at 0x13e4: file broken.cpp, line 43.
(gdb) run
Starting program: /home/tousif/Desktop/broken
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
This program is used to compute the value of the following series :
(x^0)/0! + (x^1)/1! + (x^2)/2! + (x^3)/3! + (x^4)/4! + ........ + (x^n)/n!
Please enter the value of x : 2

Please enter an integer value for n : 3


Breakpoint 1, main () at broken.cpp:43
43        double seriesValue = ComputeSeriesValue(x, n);
(gdb) s
ComputeSeriesValue (x=2, n=3) at broken.cpp:17
17        double seriesValue = 0.0;
(gdb) n
18        double xpow = 1;
(gdb) n
20        for (int k = 0; k <= n; k++) {
(gdb)
21          seriesValue += xpow / ComputeFactorial(k);
(gdb) s
```

We can see that the fact value is printed 0. Thus , the fact value is causing logical error by processing 0 evrytime.

```
21          seriesValue += xpow / ComputeFactorial(k);
(gdb) s
ComputeFactorial (number=0) at broken.cpp:7
7         int fact = 0;
(gdb)
9         for (int j = 1; j <= number; j++) {
(gdb)
13        return fact;
(gdb) print fact
$1 = 0
(gdb)
```

Thus we change the fact value as 1 in (int fact = 0;).
int fact = 1;
Then the output will be:

```
tousif@TousifVM:~$ g++ -g broken.cpp -o broken
cc1plus: fatal error: broken.cpp: No such file or directory
compilation terminated.
tousif@TousifVM:~$ cd Desktop
tousif@TousifVM:~/Desktop$ g++ -g broken.cpp -o broken
tousif@TousifVM:~/Desktop$ ./broken
This program is used to compute the value of the following series :
(x^0)/0! + (x^1)/1! + (x^2)/2! + (x^3)/3! + (x^4)/4! + ........ + (x^n)/n!
Please enter the value of x : 2

Please enter an integer value for n : 3

The value of the series for the values entered is 6.33333
tousif@TousifVM:~/Desktop$
```

2. Debugging a program that produces a core dump.

Taking testit.c files to demonstrate.

<u>testit.c</u> :-

```
#include <stdio.h>

int main()
{
        char *temp = "Paras";

        int i;
        i=0;

        temp[3]='F';

        for (i =0 ; i < 5 ; i++ )
                printf("%c\n", temp[i]);
        return 0;


}
```

<u>Process & Output</u> :-

It will throw an error “Segmentation fault”.

```
tousif@TousifVM:~$ cd Desktop
tousif@TousifVM:~/Desktop$ gcc -g testit.c -o testit
tousif@TousifVM:~/Desktop$ ./testit
Segmentation fault (core dumped)
tousif@TousifVM:~/Desktop$
```

After that, we have to generate a core file and debug it to know the cause which causing segmentation fault.

```
tousif@TousifVM:~/Desktop$ gcc -g testit.c -o testit
tousif@TousifVM:~/Desktop$ gdb testit
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
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-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 testit...
(gdb) run
Starting program: /home/tousif/Desktop/testit
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
main () at testit.c:10
10              temp[3]='F';
(gdb) generate core
warning: Memory read failed for corefile section, 4096 bytes at 0xffffffffff600000.
Saved corefile core
(gdb)
```

generate core command will generate the core file and we can confirm it with by ls.

```
tousif@TousifVM:~/Desktop$ ls
 a.out    broken.cpp   file1.txt   file3.txt
 broken   core         file2.txt   forkcall
tousif@TousifVM:~/Desktop$
```

As we can see in above debugging process , it shows error of **temp[3]='F';**.
It shows that the value **'F'** is not getting assigned to the **temp[3]** value.
If we see the declaration, **char *temp=" Paras";** it acts as string literal and we cannot modify the string literal. Thus , throwing the error.
We have to modify the declaration as character array from string literal.

char temp[]=”Paras” will be declared in place of string literal and we can get the output from this.

```
tousif@TousifVM:~/Desktop$ vi testit.c
tousif@TousifVM:~/Desktop$ gcc -g testit.c -o testit
tousif@TousifVM:~/Desktop$ ./testit
P
a
r
F
s
tousif@TousifVM:~/Desktop$
```