17/11/2020

Mohd. Aftab Alam
02013302717

EXPERIMENT 10

Aim : (a) Write a program in C/C++ to find the largest of 3 numbers. The test suite selected by a testing technique is given below-

S. No.	A	B	C	Expected O/P
1	6	10	2	10
2	10	6	2	10
3	6	2	10	10
4	6	10	20	20

Create 5 Mutants (M1 to M5) and calculate Mutation Score of this test suite.

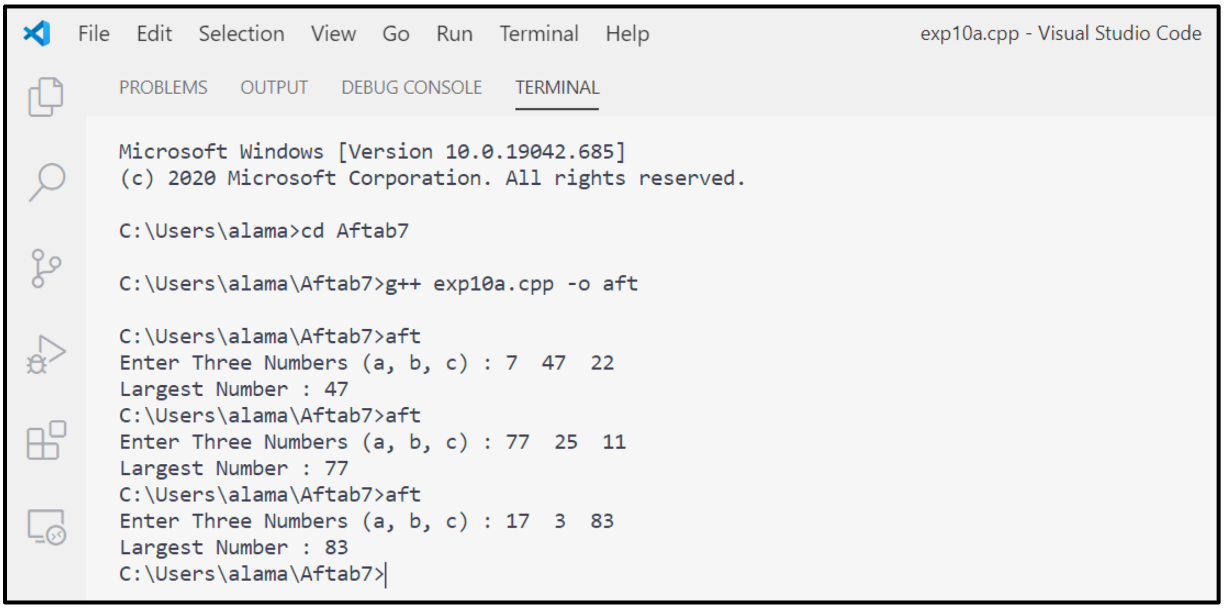
Algorithm :

- Take three numbers a, b and c as input from the user.
- If (a > = b) and (a > = c) then Largest number is a.
- Else If (b > = a) and (b > = c) then Largest number is b.
- Else the Largest number is c.

Code :

```
int main()
{
    int a, b, c;
    cout << "Enter Three Numbers (a, b, c) : ";
    cin >> a >> b >> c;
    if ((a >= b) && (a >= c))
        cout << "Largest Number : " << a;
    else if (b >= c)
        cout << "Largest Number : " << b;
    else
        cout << "Largest Number : " << c;
    return 0;
}
```

Output Screenshot :



```
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\alama>cd Aftab7

C:\Users\alama\Aftab7>g++ exp10a.cpp -o aft

C:\Users\alama\Aftab7>aft
Enter Three Numbers (a, b, c) : 7  47  22
Largest Number : 47
C:\Users\alama\Aftab7>aft
Enter Three Numbers (a, b, c) : 77  25  11
Largest Number : 77
C:\Users\alama\Aftab7>aft
Enter Three Numbers (a, b, c) : 17  3  83
Largest Number : 83
C:\Users\alama\Aftab7>
```

Mutation Testing : We will create Mutants for the above program as:

- ***Mutant 1 – M1***

```
int main()
{
int a, b, c;
cout << "Enter Three Numbers (a, b, c) : ";
cin >> a >> b >> c;
if ((c >= a) && (c <= b)) //changing the if condition in terms of c
    cout << "Largest Number : " << a;
else if (b >= c)
    cout << "Largest Number : " << b;
else
    cout << "Largest Number : " << c;
return 0;
}
```

- ***Mutant 2 – M2***

```
int main()
{
int a, b, c;
cout << "Enter Three Numbers (a, b, c) : ";
cin >> a >> b >> c;
if ((a == b) && (a >= c)) //replacing >= by ==
    cout << "Largest Number : " << a;
else if (b >= c)
    cout << "Largest Number : " << b;
else
    cout << "Largest Number : " << c;
return 0;
}
```

- ***Mutant 3 – M3***

```
int main()
{
int a, b, c;
cout << "Enter Three Numbers (a, b, c) : ";
cin >> a >> b >> c;
if ((a >= b) && (a >= c)) //replacing && by ||
    cout << "Largest Number : " << a;
else if (b >= c)
    cout << "Largest Number : " << b;
else
    cout << "Largest Number : " << c;
return 0;
}
```

- ***Mutant 4 – M4***

```
int main()
{
int a, b, c;
cout << "Enter Three Numbers (a, b, c) : ";
cin >> a >> b >> c;
if ((a >= b) && (a >= c))
    cout << "Largest Number : " << c; //replacing a by c
else if (b >= c)
    cout << "Largest Number : " << b;
else
    cout << "Largest Number : " << c;
return 0;
}
```

- ***Mutant 5 – M5***

```
int main()
{
int a, b, c;
cout << "Enter Three Numbers (a, b, c) : ";
cin >> a >> b >> c;
if ((b >= a) && (b >= c)) //changing the if condition in terms of b
    cout << "Largest Number : " << a;
else if (b >= c)
    cout << "Largest Number : " << b;
else
    cout << "Largest Number : " << c;
return 0;
}
```

Mutants Test Cases Table

Mutant	Test Case	Input	Expected Output	Mutant Output	Test Result	Remark
M1	1	<6,10,2>	10	10	Fail	Killable
	2	<10,6,2>	10	Program Terminates	Pass	Killed
	3	<6,2,10>	10	10	Fail	Killable
	4	<6,10,20>	20	20	Fail	Killable
M2	1	<6,10,2>	10	10	Fail	Killable
	2	<10,6,2>	10	Program Terminates	Pass	Killed
	3	<6,2,10>	10	10	Fail	Killable
	4	<6,10,20>	20	20	Fail	Killable
M3	1	<6,10,2>	10	Program Terminates	Pass	Killed
	2	<10,6,2>	10	10	Fail	Killable
	3	<6,2,10>	10	Program Terminates	Pass	Killed
	4	<6,10,20>	20	20	Fail	Killable
M4	1	<6,10,2>	10	10	Fail	Killable
	2	<10,6,2>	10	Program Terminates	Pass	Killed
	3	<6,2,10>	10	10	Fail	Killable
	4	<6,10,20>	20	20	Fail	Killable
M5	1	<6,10,2>	10	10	Fail	Killable
	2	<10,6,2>	10	Program Terminates	Pass	Killed
	3	<6,2,10>	10	10	Fail	Killable
	4	<6,10,20>	20	20	Fail	Killable

Mutation Score

$$\boldsymbol{Mutation\ Score} = \frac{\mathbf{100} * \boldsymbol{No.of\ Killed\ Mutants}}{(\boldsymbol{No.of\ Total\ Mutants} - \boldsymbol{No.of\ Equivalent\ Mutants})}$$

Here,

No. of Killed Mutants	=	5
No. of Total Mutants	=	5
No. of Equivalent Mutants	=	0

So,

Mutation Score	=	100 * 5 / (5 - 0)
	=	100 * 5 / 5
	=	**100**

17/11/2020 **Mohd. Aftab Alam**
02013302717

Aim : (b) Write a program in C/C++ to determine the day of the given date and perform Slice-based testing for all variables.

Algorithm :

- Take three inputs from the user for day, month, and year.
- Calculate the day of the given date using the formula:

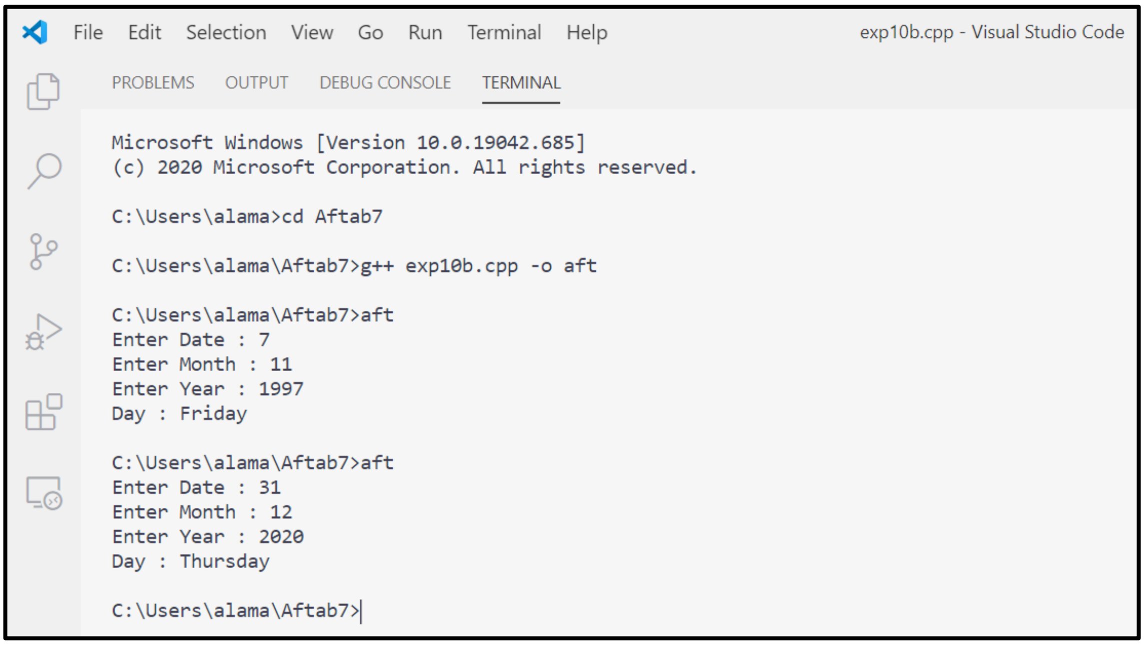
  $$Day = (d + m + y + [y / 4] + c) \bmod 7$$

- Here, c stands for the century number.
- Find the day according to the calculated number of the day.
- Print the day of the given date as calculated.

Code :

```cpp
int main()
{
int d, m, y;
cout << "Enter Date : ";
cin >> d;
cout << "Enter Month : ";
cin >> m;
cout << "Enter Year : ";
cin >> y;
const char *Names[] = {"Sunday", "Monday", "Tuesday", "Wednesday",
  "Thursday", "Friday", "Saturday"};
int day = 0;
static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};
y -= m < 3;
day = (y + y / 4 - y / 100 + y / 400 + t[m - 1] + d) % 7;
cout << "Day : " << Names[day] << endl;
return 0;
}
```

Output Screenshot :



```
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\alama>cd Aftab7

C:\Users\alama\Aftab7>g++ exp10b.cpp -o aft

C:\Users\alama\Aftab7>aft
Enter Date : 7
Enter Month : 11
Enter Year : 1997
Day : Friday

C:\Users\alama\Aftab7>aft
Enter Date : 31
Enter Month : 12
Enter Year : 2020
Day : Thursday

C:\Users\alama\Aftab7>
```

Slice Based Testing :

There is total 6 variables in the program. We can create slices for each of them.

- ***Variable: d***

S(d,5) / S(d,17) = {1-5,17}

```
1. int main()
2. {
3. int d, m, y;
4. cout << "Enter Date : ";
5. cin >> d;
17.}
```

- ***Variable: m***

S(m,7) / S(m,17) = {1-3,6,7,17}

```
1. int main()
2. {
3. int d, m, y;
6. cout << "Enter Month : ";
7. cin >> m;
17.}
```

- ***Variable: y***

S(y,9) / S(y,17) = {1-3,8,9,17}

```
1. int main()
2. {
3. int d, m, y;
8. cout << "Enter Year : ";
9. cin >> y;
17.}
```

S(y,13) / S(y,17) = {1-3,8,9,13,17}

```
1. int main()
2. {
3. int d, m, y;
8. cout << "Enter Year : ";
9. cin >> y;
13.y -= m < 3;
17.}
```

- ***Variable: Names***

S(Names,10) = {1-2,10,17}

```
1. int main()
2. {
10.const char *Names[] = {"Sunday", "Monday", "Tuesday", "Wednesday",
   "Thursday", "Friday", "Saturday"};
17.}
```

S(Names,15) / S(Names,17) = {1-17}

```
1. int main()
2. {
3. int d, m, y;
4. cout << "Enter Date : ";
5. cin >> d;
6. cout << "Enter Month : ";
7. cin >> m;
8. cout << "Enter Year : ";
9. cin >> y;
10.const char *Names[] = {"Sunday", "Monday", "Tuesday", "Wednesday",
   "Thursday", "Friday", "Saturday"};
11.int day = 0;
12.static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};
13.y -= m < 3;
14.day = (y + y / 4 - y / 100 + y / 400 + t[m - 1] + d) % 7;
15.cout << "Day : " << Names[day] << endl;
16.return 0;
17.}
```

- ***Variable: day***

S(day,11) = {1-2,11,17}

```
1. int main()
2. {
11.int day = 0;
17.}
```

S(day,14) / S(day,17) = {1-14,16,17}

```
1. int main()
2. {
3. int d, m, y;
4. cout << "Enter Date : ";
5. cin >> d;
6. cout << "Enter Month : ";
7. cin >> m;
8. cout << "Enter Year : ";
9. cin >> y;
10.const char *Names[] = {"Sunday", "Monday", "Tuesday", "Wednesday",
   "Thursday", "Friday", "Saturday"};
11.int day = 0;
12.static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};
13.y -= m < 3;
14.day = (y + y / 4 - y / 100 + y / 400 + t[m - 1] + d) % 7;
16.return 0;
17.}
```

- ***Variable: t***

S(t,12) / S(t,17) = {1-2,12,17}

```
1. int main()
2. {
12.static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};
17.}
```

Test Cases :

S. No.	Slice	Lines Covered	Variables			Expected Output
			d	m	y	
1	S(d,5) / S(d,17)	1-5,17	12			No Output
2	S(m,7) / S(m,17)	1-3,6,7,17		4		No Output
3	S(y,9)	1-3,8,9,17			2019	No Output
4	S(y,13) / S(y,17)	1-3,8-9,1317			2019	No Output
5	S(Names,10)	1-2,10,17				No Output
6	S(Names,15) / S(Names,17)	1-17	12	4	2019	Friday
7	S(day,11)	1-2,11,17				No Output
8	S(day,14) / S(day,17)	1-14,16,17	12	4	2019	No Output
9	S(t,12) / S(t,17)	1,2,12,17				No Output