[CS200]- Software Tool and Technologies Lab-II **Fall 2020-21**

Homework: IV

Team Name: Peaky Blinders Roll Numbers:11940150, 11940370, 11940380

Solution of Problem 1:

(a) Using **gdb** commands, the values of the variables X, result, answer, and y at the following lines in the program: 7, 10, 12, 17, 21, 23 are recorded and reported in the log file- "**gdb1ac**".

Following **gdb commands** are used for the given question:-

b main //Set a Breakpoint at the beginning of the main function or line-5
r //Start running the program with no command line argument
p X //Print the value of a variable X at line-7
p result //Print the value of a variable result at line-7
p answer //Print the value of a variable answer at line-7
p y //Print the value of a variable y at line-7
n //Next command that is Step-Over the function
n //Next command again and again upto line-10
p X //Print the value of a variable X at line-10 and Similarly print the value of all remaining variables at line-10
n //Next command that is Step-Over the function
s //Step command that is just single Step into the function if the statement is a function call
// Similarly all the 4 variables are printed at each of the lines asked in the question which can be seen in the log file- "**gdb1ac**"

(b) The value of result in line 10 is **6422352**.This has been shown in the log file- "**gdb1b**".

In any program, if a variable is allocated but not assigned or initialized with any value then it can be said that it contains a "**garbage value**". Actually, allocation of a variable means to reserve some memory for that variable and this "garbage value" is the information that is held by any random location in the computer's memory. Garbage value is a value that the memory location which the variable is assigned already has **due its use in some earlier other program.**

Example-

```
int v;
```

Here, 'v' is an integer type variable but it has not been assigned by any value. So, it automatically gets a garbage value by default which can be any value.

If we run the program several times, the value of result at line-10 **does not change**. This has been shown in the log file- "**gdb1b**". It remains same when we ran it multiple times but in C++ trying to print the value of variable which has not been assigned any value gives "**undefined behaviour**" so the there is not particular reason for the value staying same.

(c) The values printed for X on line-10 is '**10**' and on lines-17 is '**30**'. This has been shown in the log file- "**gdb1ac**".

The value of X at line-10 is '10' because the variable X has been initialized with the value '10' in line no-9 but in the case of the value of variable X at line-17, its value is '30' rather then '5'(i.e. the global value of the variable X defined at line no-4). Actually, this is because a local variable may have the wrong value at certain points in a function—just after the entry to a new scope, and before the exit.
When we are stepping by machine instructions, the above mentioned problem occurs because on most of the machines it takes multiple instructions to set up a stack frame that includes the local variable definitions. If we are stepping by machine instructions then the variables may have the wrong values until the stack frame is completely built.

This same reason can be applied on the value of vaiable X at the line no-7(i.e. = 4201120).

Reference- *https://sourceware.org/gdb/current/onlinedocs/gdb/Variables.html*