

## Debugging Assignment

Q1.

```
#include <stdio.h>

int main() {

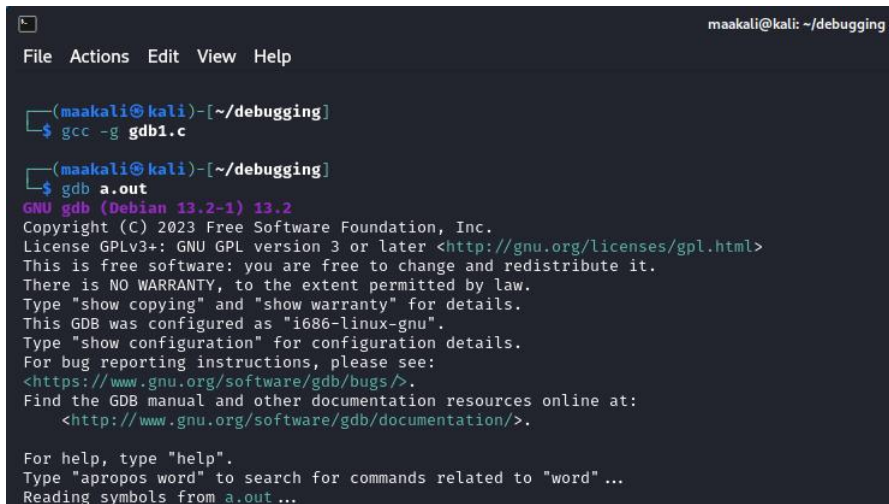
    int a,b;

    int c=a+b;

    printf("%d",c); // Output: 5

    return 0;

}
```

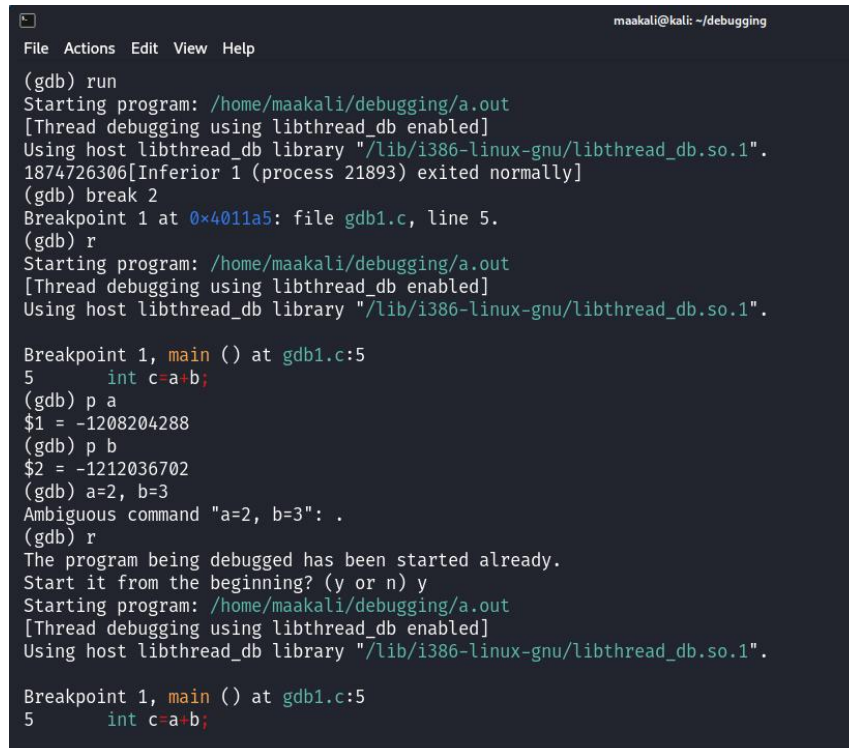


```
maakali@kali: ~/debugging
File Actions Edit View Help

(maakali@kali)-[~/debugging]
$ gcc -g gdb1.c

(maakali@kali)-[~/debugging]
$ gdb a.out
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "i686-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 a.out ...
```



```
File Actions Edit View Help

(gdb) run
Starting program: /home/maakali/debugging/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".
1874726306[Inferior 1 (process 21893) exited normally]
(gdb) break 2
Breakpoint 1 at 0x4011a5: file gdb1.c, line 5.
(gdb) r
Starting program: /home/maakali/debugging/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb1.c:5
5       int c=a+b;
(gdb) p a
$1 = -1208204288
(gdb) p b
$2 = -1212036702
(gdb) a=2, b=3
Ambiguous command "a=2, b=3": .
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/maakali/debugging/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

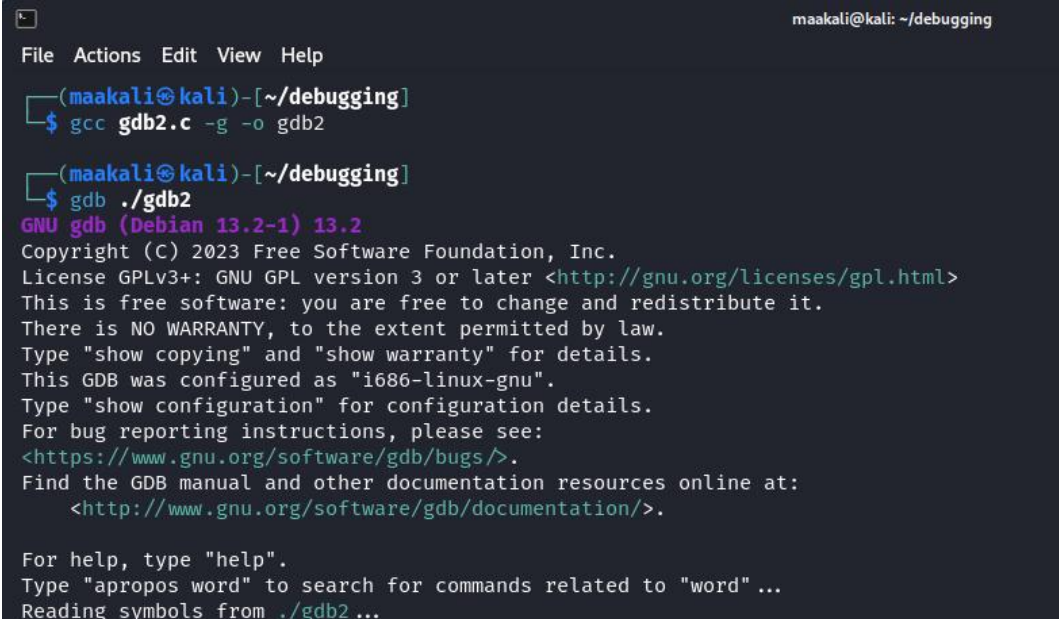
Breakpoint 1, main () at gdb1.c:5
5       int c=a+b;
```

Q.2)

```
#include int main() {  
int d=2;  
printf("Enter the value of d:");  
scanf("%d",d);  
printf("The value of d is:%d",d);  
return 0;  
}
```

### Correct code

```
#include <stdio.h>  
int main()  
{  
int d=2;  
printf("Enter the value of d:");  
scanf("%d",&d);  
printf("The value of d is:%d",d);  
return 0;  
}
```



```
maakali@kali: ~/debugging  
File Actions Edit View Help  
(maakali@kali)-[~/debugging]  
$ gcc gdb2.c -g -o gdb2  
(maakali@kali)-[~/debugging]  
$ gdb ./gdb2  
GNU gdb (Debian 13.2-1) 13.2  
Copyright (C) 2023 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 "i686-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 ./gdb2...
```

```
maakali@kali: ~/debugging
File Actions Edit View Help
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./gdb2...
(gdb) list
1      #include <stdio.h>
2      int main()
3      {
4          int d=2;
5          printf("Enter the value of d:");
6          scanf("%d",d);
7          printf("The value of d is:%d",d);
8          return 0;
9      }
(gdb) break main
Breakpoint 1 at 0x11b6: file gdb2.c, line 4.
(gdb) r
Starting program: /home/maakali/debugging/gdb2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb2.c:4
4      int d=2;
(gdb) n
5      printf("Enter the value of d:");
(gdb) n
6      scanf("%d",d);
(gdb) n
Enter the value of d:7

Program received signal SIGSEGV, Segmentation fault.
0xb7c61e92 in __vfscanf_internal (s=<optimized out>, format=<optimized out>, argptr=<optimized out>, mode_flags=<optimized out>)
    at ./stdio-common/vfscanf-internal.c:1896
1896 _./stdio-common/vfscanf-internal.c: No such file or directory.
```

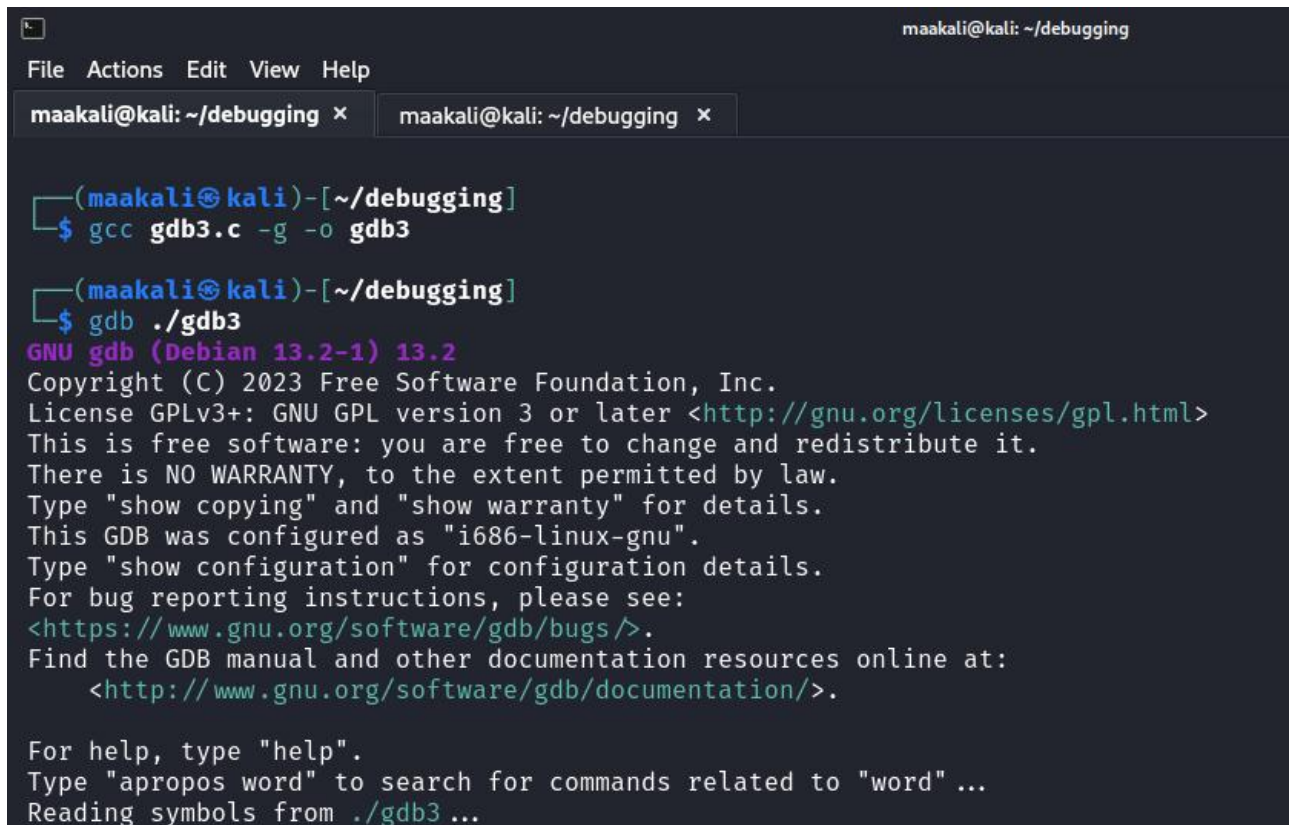
Q.3)

```
#include <stdio.h>
#include <stdlib.h>
int factorial(int n);
int main(void) {
    int n = 5;
    int f = factorial(n);
    printf("The factorial of %d is %d.\n", n, f); n = 17;
    f = factorial(n);
    printf("The factorial of %d is %d.\n", n, f); return 0;
}
//A factorial is calculated by  $n! = n * (n - 1) * (n - 2) * \dots * 1$  //E.g.  $5! = 5 * 4 * 3 * 2 * 1 = 120$ 
int factorial(int n)
{ int f = 1; int i = 1;
while (i <= n) { f = f * i; i++; }
return f;
}
```

### Correct code Q.3

```
#include <stdio.h>
#include <stdlib.h>
long long int factorial(int n);
int main(void) {
    int n = 5;
    long long int f = factorial(n);
    printf("The factorial of %d is %lld.\n", n, f);
    n = 17;
    f = factorial(n);
    printf("The factorial of %d is %lld.\n", n, f);
    return 0;
}
```

```
//A factorial is calculated by  $n! = n * (n - 1) * (n - 2) * \dots * 1$   
//E.g.  $5! = 5 * 4 * 3 * 2 * 1 = 120$   
long long int factorial(int n) {  
    long long int f = 1;  
    int i = 1;  
    while (i <= n) {  
        f = f * i;  
        i++;  
    }  
    return f;  
}
```



The screenshot shows a terminal window with a dark background. At the top, there's a title bar with a window icon and the text 'maakali@kali: ~/debugging'. Below the title bar is a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Underneath the menu bar are two tabs, both labeled 'maakali@kali: ~/debugging' with a close button 'x'. The terminal content shows the user at the prompt '(maakali@kali)-[~/debugging]' typing the command '\$ gcc gdb3.c -g -o gdb3'. The next prompt shows '\$ gdb ./gdb3', followed by the GDB startup banner for version 13.2, including copyright information and links to the GNU website. The banner ends with 'Reading symbols from ./gdb3 ...'.

```
(maakali@kali)-[~/debugging]  
$ gcc gdb3.c -g -o gdb3  
  
(maakali@kali)-[~/debugging]  
$ gdb ./gdb3  
GNU gdb (Debian 13.2-1) 13.2  
Copyright (C) 2023 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 "i686-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 ./gdb3 ...
```

```

maakali@kali: ~/debugging
File Actions Edit View Help
maakali@kali: ~/debugging x maakali@kali: ~/debugging x

(gdb) run
Starting program: /home/maakali/debugging/gdb3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".
The factorial of 5 is 120.
The factorial of 17 is -288522240.
[Inferior 1 (process 23612) exited normally]
(gdb) break main
Breakpoint 1 at 0x4011a6: file gdb3.c, line 5.
(gdb) r
Starting program: /home/maakali/debugging/gdb3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb3.c:5
5      int n = 5;
(gdb) n
6      int f = factorial(n);
(gdb) n
7      printf("The factorial of %d is %d.\n", n, f);
(gdb) n
The factorial of 5 is 120.
8      n = 17;
(gdb) n
9      f = factorial(n);
(gdb) n
10     printf("The factorial of %d is %d.\n", n, f);
(gdb) n
The factorial of 17 is -288522240.
11     return 0;
(gdb) break factorial(n)
Function "factorial(n)" not defined.

```

```

maakali@kali: ~/debugging
File Actions Edit View Help
maakali@kali: ~/debugging x maakali@kali: ~/debugging x

Breakpoint 3 at 0x401225: file gdb3.c, line 16.
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/maakali/debugging/gdb3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb3.c:5
5      int n = 5;
(gdb) n
6      int f = factorial(n);
(gdb) n

Breakpoint 3, factorial (n=5) at gdb3.c:16
16     int f = 1;
(gdb) n
17     int i = 1;
(gdb) n
18     while (i <= n) {
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) c
Continuing.
The factorial of 5 is 120.

Breakpoint 3, factorial (n=17) at gdb3.c:16
16     int f = 1;
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) c
Continuing.
The factorial of 17 is -288522240.
[Inferior 1 (process 24880) exited normally]

```

**Q.4)**

```
#include <stdio.h>

int main(void) {
    int arr[2];
    arr[3] = 10; // Accessing out of bound
    return (0);
}
```

Correct code for Q.4):-

```
#include <stdio.h>

int main(void) {
    int arr[4];
    arr[3] = 10;
    return (0);
}
```



```

maakali@kali: ~/de
File Actions Edit View Help

(maakali@kali)-[~/debugging]
$ gcc gdb4.c -g -o gdb4

(maakali@kali)-[~/debugging]
$ gdb ./gdb4
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "i686-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 ./gdb4 ...

```

```

maakali@kali: ~/debugging
File Actions Edit View Help

Reading symbols from ./gdb4 ...
(gdb) list
1      #include <stdio.h>
2      int main(void)
3      {
4          int arr[2];
5          arr[3] = 10; // Accessing out of bound
6          return (0);
7      }
8
9
(gdb) break main
Breakpoint 1 at 0x1189: file gdb4.c, line 5.
(gdb) r
Starting program: /home/maakali/debugging/gdb4
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb4.c:5
5      arr[3] = 10; // Accessing out of bound
(gdb) print arr[3]
$1 = -1212008507
(gdb) n
6      return (0);
(gdb) p arr[3]
$2 = 10
(gdb) continue
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x0000000a in ?? ()
(gdb) quit
A debugging session is active.

        Inferior 1 [process 1293] will be killed.

Quit anyway? (y or n) y

```



**Q.5)**

```
#include <stdio.h>

int main(){

int *p;

printf("%d",*p);

return 0;

}
```

Correct code Q.5):-

```
#include <stdio.h>

int main()

{

int *p;

int var = 15;

p = &var;

printf("%d",*p);

return 0;

}
```

```

maakali@kali: ~/debugging
File Actions Edit View Help

(maakali@kali)-[~/debugging]
$ gcc gdb5.c

(maakali@kali)-[~/debugging]
$ ./a.out
-1212153856

(maakali@kali)-[~/debugging]
$ gcc gdb5.c -g -o gdb5

(maakali@kali)-[~/debugging]
$ gdb ./gdb5
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "i686-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 ./gdb5 ...

```

```

maakali@kali: ~/debugging
File Actions Edit View Help

$ gdb ./gdb5
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "i686-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 ./gdb5 ...
(gdb) break main
Breakpoint 1 at 0x11a5: file gdb5.c, line 5.
(gdb) run
Starting program: /home/maakali/debugging/gdb5
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at gdb5.c:5
5      printf("%d",*p);
(gdb) next
6      return 0;
(gdb) print *p
$p1 = -1212153856
(gdb) continue
Continuing.
-1212153856[Inferior 1 (process 8193) exited normally]

```