

## Exercise 1

```
m4eeb@id414m40:2263 lab4$ gcc -g ex1.c -o ex1
m4eeb@id414m40:2263 lab4$ gdb ./ex1
GNU gdb (GDB) Rocky Linux 10.2-10.el9.0.1
Copyright (C) 2021 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-redhat-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 ./ex1...
(gdb) break dummy_frame
Breakpoint 1 at 0x40112a: file ex1.c, line 13.
(gdb) run
Starting program: /home1/ugrads/m4eeb/Desktop/2263 lab4/ex1
(Thread debugging using libthread_db enabled)
Using host libthread_db library "/lib64/libthread_db.so.1".
main: a = 1 0x7fffffffdf30
main: a = 2 0x7fffffffdf34
main: a = 3 0x7fffffffdf38
main: a = 4 0x7fffffffdf3c
main: a = 5 0x7fffffffdf40
Breakpoint 1, dummy_frame () at ex1.c:13
13      return;
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.34-60.el9.2.7.x86_64
(gdb) bt
#0  dummy_frame () at ex1.c:13
#1  0x00007fffffd130 in main (argc=1, argv=0x7fffffffef088) at ex1.c:31
(gdb) info frame #
Stack frame at 0x7fffffffdf40:
rip = 0x40112a in dummy_frame (ex1.c:13); saved rip = 0x4011a6
called by frame at 0x7fffffffdf30
source language c
Arglist at 0x7fffffffdf30, args:
Locals at 0x7fffffffdf30, Previous frame's sp is 0x7fffffffdf40
Saved registers:
_rbp at 0x7fffffffdf30, rip at 0x7fffffffdf30
(gdb) exit
Undefined command: "exit". Try "help".
(gdb) q
A debugging session is active.

Inferior 1 [process 1283928] will be killed.

Quit anyway? (y or n) y
m4eeb@id414m40:2263 lab4$
```

## 1.2

```
#include <stdio.h>
#include <stdlib.h>

void dummy_frame()
{
    return;
}

int main(int argc, char * * argv)
{
    int i;
    int *a = (int *)malloc(5 * sizeof(int));
    if (a == NULL) {
        fprintf(stderr, "Memory allocation failed\n");
        return EXIT_FAILURE;
    }

    for (i = 0; i < 5; i++) {
        a[i] = i + 1;
        printf("main: a[%d] = %d %p \n", i, a[i], &a[i]);
    }
}
```

```

dummy_frame();

free(a);
return EXIT_SUCCESS;
}

```

```

m4eeb@id414m40:2263 lab4
[m4eeb@id414m40 2263 lab4]$ gcc -g ex1_modified.c -o ex1_modified
[m4eeb@id414m40 2263 lab4]$ gdb ./ex1_modified
GNU gdb (GDB) Rocky Linux 10.2-10.el9.0.1
Copyright (C) 2021 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-redhat-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 ./ex1_modified...
(gdb) break dummy_frame
Breakpoint 1 at 0x401221: file ex1_modified.c, line 7.
(gdb) run
Starting program: /home1/ugrads/m4eeb/Desktop/2263 lab4/ex1_modified
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
main: a[0] = 1 0x4052a0
main: a[1] = 2 0x4052a4
main: a[2] = 3 0x4052a8
main: a[3] = 4 0x4052ac
main: a[4] = 5 0x4052b0

Breakpoint 1, dummy_frame () at ex1_modified.c:7
7       return;
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.34-60.el9.2.7.x86_64
(gdb) bt
#0  dummy_frame () at ex1_modified.c:7
#1  0x0000000000000000 in main (argc=1, argv=0x7fffffffef078)
    at ex1_modified.c:24
(gdb) info frame 0
Stack frame at 0x7fffffffdf40:
rip = 0x401221 in dummy_frame (ex1_modified.c:7); saved rip = 0x401221
called by frame at 0x7fffffffdf70
source language c.
Arglist at 0x7fffffffdf30, args:
Locals at 0x7fffffffdf30, Previous frame's sp is 0x7fffffffdf40
Saved registers:
rip at 0x7fffffffdf30, rip at 0x7fffffffdf30
(gdb) q
A debugging session is active.

Inferior 1 [process 1285627] will be killed.

Quit anyway? (y or n) y
[m4eeb@id414m40 2263 lab4]$

```

No, the addresses of array elements are not falling within the range of the main function frame. This is because when the array `a[ ]` is allocated on the heap using `malloc()`, it is not stored on the stack within the main function's frame. Instead, it is allocated dynamically on the heap, which is separate from the stack frame of the main function. Therefore, the addresses of array elements are not within the range of the main function frame.

## Exercise 2

```
#include <stdio.h>
#include <stdlib.h>

void dummy_frame()
{
    return;
}

int main(int argc, char * * argv)
{
    int i;
    int *a = (int *)malloc(5 * sizeof(int));
    if (a == NULL) {
        fprintf(stderr, "Memory allocation failed\n");
        return EXIT_FAILURE;
    }

    printf("Original array:\n");
    for (i = 0; i < 5; i++) {
        a[i] = i + 1;
        printf("main: a[%d] = %d %p \n", i, a[i], &a[i]);
    }

    int *temp = (int *)realloc(a, 8 * sizeof(int));
    if (temp == NULL) {
        fprintf(stderr, "Memory reallocation failed\n");
        free(a);
        return EXIT_FAILURE;
    } else {
        a = temp;
    }

    for (i = 5; i < 8; i++) {
        a[i] = i + 1;
    }

    printf("\nExtended array:\n");
    for (i = 0; i < 8; i++) {
        printf("main: a[%d] = %d %p \n", i, a[i], &a[i]);
    }

    free(a);
    return EXIT_SUCCESS;
}
```

}

```
m4eeb@id414m40:2263 lab4$ gcc -o ex1_extended ex1_extended.c
[m4eeb@id414m40:2263 lab4]$ ./ex1_extended
Original array:
main: a[0] = 1 0x23462a0
main: a[1] = 2 0x23462a4
main: a[2] = 3 0x23462a8
main: a[3] = 4 0x23462ac
main: a[4] = 5 0x23462b0
Extended array:
main: a[0] = 1 0x23466d0
main: a[1] = 2 0x23466d4
main: a[2] = 3 0x23466d8
main: a[3] = 4 0x23466dc
main: a[4] = 5 0x23466e0
main: a[5] = 6 0x23466e4
main: a[6] = 7 0x23466e8
main: a[7] = 8 0x23466ec
[m4eeb@id414m40:2263 lab4]$
```

The addresses for the new extended array may not be the same as the original array. When you extend the array using `realloc()`, the memory allocator might allocate a new memory block and copy the contents of the original array to the new memory block. In such a case, the addresses of the elements in the extended array would be different from the original array. However, if `realloc()` can extend the original memory block without moving it, the addresses might remain the same. This behavior depends on various factors like the availability of contiguous memory, system memory management strategies, etc.

## Exercise 3

```
m4eeb@remotelabm42:2263 lab4
File Edit View Search Terminal Help
[m4eeb@remotelabm42 2263 lab4]$ gcc -o ex1_extended_no_free ex1_extended_no_free.c
[m4eeb@remotelabm42 2263 lab4]$ valgrind ./ex1_extended_no_free
==4103395== Memcheck, a memory error detector
==4103395== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==4103395== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright info
==4103395== Command: ./ex1_extended_no_free
==4103395==
Original array:
main: a[0] = 1 0x4a72040
main: a[1] = 2 0x4a72044
main: a[2] = 3 0x4a72048
main: a[3] = 4 0x4a7204c
main: a[4] = 5 0x4a72050

Extended array:
main: a[0] = 1 0x4a724e0
main: a[1] = 2 0x4a724e4
main: a[2] = 3 0x4a724e8
main: a[3] = 4 0x4a724ec
main: a[4] = 5 0x4a724f0
main: a[5] = 6 0x4a724f4
main: a[6] = 7 0x4a724f8
main: a[7] = 8 0x4a724fc
==4103395==
==4103395== HEAP SUMMARY:
==4103395==    in use at exit: 32 bytes in 1 blocks
==4103395== total heap usage: 3 allocs, 2 frees, 1,076 bytes allocated
==4103395==
==4103395== LEAK SUMMARY:
==4103395==    definitely lost: 32 bytes in 1 blocks
==4103395==    indirectly lost: 0 bytes in 0 blocks
==4103395==    possibly lost: 0 bytes in 0 blocks
==4103395==    still reachable: 0 bytes in 0 blocks
==4103395==    suppressed: 0 bytes in 0 blocks
==4103395== Rerun with --leak-check=full to see details of leaked memory
==4103395==
==4103395== For lists of detected and suppressed errors, rerun with: -s
==4103395== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[m4eeb@remotelabm42 2263 lab4]$
```

## 3.2

### #source code

```
#include <stdio.h>
#include <stdlib.h>

void dummy_frame()
{
    return;
}

int main(int argc, char * * argv)
{
    int i;
    int *a = (int *)malloc(5 * sizeof(int));
    if (a == NULL) {
        fprintf(stderr, "Memory allocation failed\n");
        return EXIT_FAILURE;
    }

    printf("Original array:\n");
    for (i = 0; i < 5; i++) {
        a[i] = i + 1;
        printf("main: a[%d] = %d %p \n", i, a[i], &a[i]);
    }

    int *temp = (int *)realloc(a, 8 * sizeof(int));
    if (temp == NULL) {
        fprintf(stderr, "Memory reallocation failed\n");
        free(a);
        return EXIT_FAILURE;
    } else {
        a = temp;
    }

    for (i = 5; i < 8; i++) {
        a[i] = i + 1;
    }

    printf("\nExtended array:\n");
    for (i = 0; i < 8; i++) {
        printf("main: a[%d] = %d %p \n", i, a[i], &a[i]);
    }
}
```

```

free(a);

return EXIT_SUCCESS;
}

```

```

m4eeb@remotelabm42:2263 lab4
File Edit View Search Terminal Help
[m4eeb@remotelabm42 2263 lab4]$ gcc -g ex1_extended_fixed.c ./ex1_extended_fixed
/usr/bin/ld: cannot find ./ex1_extended_fixed: No such file or directory
collect2: error: ld returned 1 exit status
[m4eeb@remotelabm42 2263 lab4]$ gcc -g ex1_extended_fixed.c -o ex1_extended_fixed
[m4eeb@remotelabm42 2263 lab4]$ valgrind ./ex1_extended_fixed
==4107212== Memcheck, a memory error detector
==4107212== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==4107212== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright info
==4107212== Command: ./ex1_extended_fixed
==4107212==
Original array:
main: a[0] = 1 0x4a72040
main: a[1] = 2 0x4a72044
main: a[2] = 3 0x4a72048
main: a[3] = 4 0x4a7204c
main: a[4] = 5 0x4a72050

Extended array:
main: a[0] = 1 0x4a724e0
main: a[1] = 2 0x4a724e4
main: a[2] = 3 0x4a724e8
main: a[3] = 4 0x4a724ec
main: a[4] = 5 0x4a724f0
main: a[5] = 6 0x4a724f4
main: a[6] = 7 0x4a724f8
main: a[7] = 8 0x4a724fc
==4107212==
==4107212== HEAP SUMMARY:
==4107212==      in use at exit: 0 bytes in 0 blocks
==4107212==    total heap usage: 3 allocs, 3 frees, 1,076 bytes allocated
==4107212==
==4107212== All heap blocks were freed -- no leaks are possible
==4107212==
==4107212== For lists of detected and suppressed errors, rerun with: -s
==4107212== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[m4eeb@remotelabm42 2263 lab4]$

```