



Regras:

1. Se este template não for submetido e preenchido corretamente, será atribuída a cotação zero ao laboratório;
2. Não se esqueça de submeter a pasta com o projeto do laboratório (código-fonte + makefile); sem código é atribuído zero ao laboratório.
3. Não "falsifique" os outputs dos níveis; alguma situação dessas resultará na atribuição de zero a dois laboratórios consecutivos.

Identificação Alunos

- 190221093 | Alexandre Coelho
- 190221128 | Sérgio Veríssimo

Nível 1 – Código de teste + output

```
int *array;
int n;
printf("Length of fib sequence?: ");
scanf("%d",&n);
array = fibArrayCreate(n);
printf("Address of fib array: %p\n",&array);
fibArrayPrint(array,n);
free(array);
printf("Address of fib array: %p\n",&array);
return EXIT_SUCCESS;
```

Output:

```
==13986== Memcheck, a memory error detector
==13986== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==13986== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==13986== Command: ./prog
==13986==
==13986== error calling PR_SET_PTRACER, vgdb might block
Length of fib sequence?:10
Address of fib array: 0x1ffefffb30
{ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34}
Address of fib array: 0x1ffefffb30
==13986==
==13986== HEAP SUMMARY:
```

```

==13986==   in use at exit: 0 bytes in 0 blocks
==13986== total heap usage: 3 allocs, 3 frees, 2,088 bytes allocated
==13986==
==13986== All heap blocks were freed -- no leaks are possible
==13986==
==13986== For counts of detected and suppressed errors, rerun with: -v
==13986== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Nível 2 – Código de teste + output

```

int *array;
int n;
int *newArray;

printf("Length of fib sequence?:");
scanf("%d",&n);
array = fibArrayCreate(n);
printf("Address of fib array: %p\n",&array);
fibArrayPrint(array,n);

newArray = fibArrayCopy(array,n);
printf("Address of fib array copy: %p\n",&newArray);
fibArrayPrint(newArray,n);

free(array);
printf("Address of fib array: %p\n",&array);
free(newArray);

```

Output:

```

==222== Memcheck, a memory error detector
==222== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==222== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==222== Command: ./prog
==222==
==222== error calling PR_SET_PTRACER, vgdb might block
Length of fib sequence?:5
Address of fib array: 0x1ffefffb58
{ 0, 1, 1, 2, 3}
Address of fib array copy: 0x1ffefffb60
{ 0, 1, 1, 2, 3}
Address of fib array: 0x1ffefffb58
==222==
==222== HEAP SUMMARY:
==222==   in use at exit: 0 bytes in 0 blocks
==222== total heap usage: 4 allocs, 4 frees, 2,088 bytes allocated
==222==
==222== All heap blocks were freed -- no leaks are possible

```

```

==222==
==222== For counts of detected and suppressed errors, rerun with: -v
==222== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Nível 3 – Código de teste + output

```

int *array;
int **arrayPointer = &array;
int n;

printf("Length of fib sequence?:");
scanf("%d", &n);
array = fibArrayCreate(n);
printf("Address of fib array: %p\n", &array);
fibArrayPrint(array,n);
fibArrayDestroy(arrayPointer);
printf("Address of fib array: %p", *arrayPointer);
fibArrayPrint(array,n);

```

Output:

```

==1945== Memcheck, a memory error detector
==1945== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==1945== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==1945== Command: ./prog
==1945==
==1945== error calling PR_SET_PTRACER, vgdb might block
Length of fib sequence?:6
Address of fib array: 0x1ffefffb58
{ 0, 1, 1, 2, 3, 5}
Address of fib array: (nil)
(NULL)
==1945==
==1945== HEAP SUMMARY:
==1945==   in use at exit: 0 bytes in 0 blocks
==1945== total heap usage: 3 allocs, 3 frees, 2,072 bytes allocated
==1945==
==1945== All heap blocks were freed -- no leaks are possible
==1945==
==1945== For counts of detected and suppressed errors, rerun with: -v
==1945== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Nível 4 – Código de teste + output

```

int *array;
int **arrayPointer = &array;
int n;

```

```

printf("Length of fib sequence?: ");
scanf("%d", &n);
array = fibArrayCreate(n);
printf("Address of fib array: %p\n", &array);
fibArrayPrint(array,n);
printf("Address of fib array: %p", &array);
fibArrayExpand(arrayPointer, &n);
fibArrayPrint(array,n);
fibArrayDestroy(arrayPointer);

```

Output:

```

==935== Memcheck, a memory error detector
==935== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==935== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==935== Command: ./prog
==935==
==935== error calling PR_SET_PTRACER, vgdb might block
Length of fib sequence?: 6
Address of fib array: 0x1 ffeffb58
{ 0, 1, 1, 2, 3, 5}
Address of fib array: 0x1 ffeffb58
{ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89}
==935==
==935== HEAP SUMMARY:
==935==   in use at exit: 0 bytes in 0 blocks
==935== total heap usage: 4 allocs, 4 frees, 2,120 bytes allocated
==935==
==935== All heap blocks were freed -- no leaks are possible
==935==
==935== For counts of detected and suppressed errors, rerun with: -v
==935== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Nível 5 – Código de teste + output

```

int n;

printf("Length of fib sequence?: ");
scanf("%d", &n);
PtArray pointer = fibArrayCreate(n);
printf("Address of fib array: %p\n", &pointer);
fibArrayPrint(pointer);
printf("Address of fib array: %p\n", &pointer);
fibArrayExpand(pointer);
fibArrayPrint(pointer);
fibArrayDestroy(&pointer);

```

Output:

```

==15827== Memcheck, a memory error detector
==15827== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.

```

```
==15827== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==15827== Command: ./prog
==15827==
==15827== error calling PR_SET_PTRACER, vgdb might block
Length of fib sequence?: 6
New: 0x522d8c0
Address of fib array: 0x1ffefffb30
{ 0, 1, 1, 2, 3, 5}
Address of fib array: 0x1ffefffb30
{ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89}
==15827==
==15827== HEAP SUMMARY:
==15827==   in use at exit: 0 bytes in 0 blocks
==15827== total heap usage: 5 allocs, 5 frees, 2,136 bytes allocated
==15827==
==15827== All heap blocks were freed -- no leaks are possible
==15827==
==15827== For counts of detected and suppressed errors, rerun with: -v
==15827== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```
