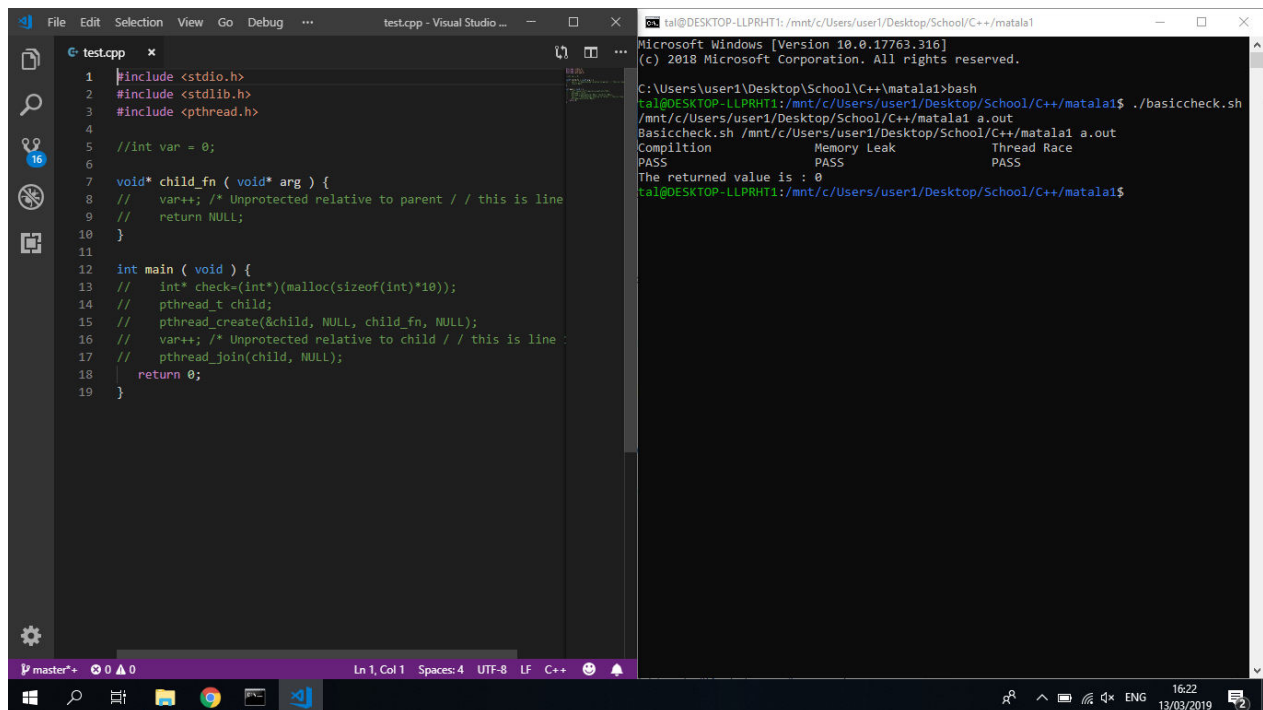


# קובץ דוגמאות הרצה

דוגמא 1: ריצה מוצלחת.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4
5 //int var = 0;
6
7 void* child_fn ( void* arg ) {
8     // var++; /* Unprotected relative to parent / / this is line
9     // return NULL;
10 }
11
12 int main ( void ) {
13     // int* check=(int*)(malloc(sizeof(int)*10));
14     // pthread_t child;
15     // pthread_create(&child, NULL, child_fn, NULL);
16     // var++; /* Unprotected relative to child / / this is line
17     // pthread_join(child, NULL);
18     return 0;
19 }
```

```
Microsoft Windows [Version 10.0.17763.316]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user1\Desktop\School\C++\matala1>bash
tal@DESKTOP-LLPRHT1:/mnt/c/Users/user1/Desktop/School/C++/matala1$ ./basiccheck.sh
/mnt/c/Users/user1/Desktop/School/C++/matala1 a.out
Basiccheck.sh /mnt/c/Users/user1/Desktop/School/C++/matala1 a.out
Compilation      Memory Leak      Thread Race
PASS             PASS             PASS
The returned value is : 0
tal@DESKTOP-LLPRHT1:/mnt/c/Users/user1/Desktop/School/C++/matala1$
```

## דוגמא 2: Memory leak

The screenshot shows a Visual Studio window with a C++ file named `test.cpp`. The code defines a child function `child_fn` that increments a global variable `var` and returns `NULL`. The `main` function allocates memory for an array `check`, creates a child thread, and joins it. The output window shows the results of a `basiccheck.sh` script, which reports a memory leak failure.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4
5 //int var = 0;
6
7 void* child_fn ( void* arg ) {
8     // var++; /* Unprotected relative to parent / / this is line
9     // return NULL;
10 }
11
12 int main ( void ) {
13     int* check=(int*)(malloc(sizeof(int)*10));
14     pthread_t child;
15     pthread_create(&child, NULL, child_fn, NULL);
16     // var++; /* Unprotected relative to child / / this is line
17     pthread_join(child, NULL);
18     return 0;
19 }
```

```
tal@DESKTOP-LLPRHT1: /mnt/c/Users/user1/Desktop/School/C++/matala1
./basiccheck.sh
Basiccheck.sh /mnt/c/Users/user1/Desktop/School/C++/matala1 a.out
Compilation      Memory Leak      Thread Race
PASS             FAIL             PASS
The returned value is : 2
tal@DESKTOP-LLPRHT1: /mnt/c/Users/user1/Desktop/School/C++/matala1$
```

## דוגמא 3: Thread race

The screenshot shows a Visual Studio window with a C++ file named `test.cpp`. The code is similar to the previous example, but the `main` function includes a string `asdsadasdsa` after allocating the `check` array. The output window shows the results of a `basiccheck.sh` script, which reports a thread race failure.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4
5 //int var = 0;
6
7 void* child_fn ( void* arg ) {
8     // var++; /* Unprotected relative to parent / / this is line
9     // return NULL;
10 }
11
12 int main ( void ) {
13     int* check=(int*)(malloc(sizeof(int)*10));asdsadasdsa
14     pthread_t child;
15     pthread_create(&child, NULL, child_fn, NULL);
16     // var++; /* Unprotected relative to child / / this is line
17     pthread_join(child, NULL);
18     return 0;
19 }
```

```
tal@DESKTOP-LLPRHT1: /mnt/c/Users/user1/Desktop/School/C++/matala1
./basiccheck.sh
Basiccheck.sh /mnt/c/Users/user1/Desktop/School/C++/matala1 a.out
Compilation      Memory Leak      Thread Race
FAIL             FAIL             FAIL
tal@DESKTOP-LLPRHT1: /mnt/c/Users/user1/Desktop/School/C++/matala1$
```

## דוגמא 4: קומפילציה שנכשלה

The image shows a screenshot of a development environment. On the left, the Visual Studio code editor displays a C++ file named `test.cpp`. The code includes `<stdio.h>`, `<stdlib.h>`, and `<pthread.h>`. It defines a global variable `int var = 0;` and a function `void* child_fn ( void* arg )` that increments `var` and returns `NULL`. The `main` function creates a child thread using `pthread_create` and joins it with `pthread_join`. Comments indicate that the child thread increments `var` (line 6) and the main thread increments `var` (line 13). The status bar at the bottom shows 'Ln 5, Col 1', 'Spaces: 4', 'UTF-8', 'LF', and 'C++'.

On the right, a terminal window shows the execution of the program. The user runs `./basiccheck.sh`, which outputs the following results:

```
tal@DESKTOP-LLPRHT1: /mnt/c/Users/user1/Desktop/School/C++/matala1$ ./basiccheck.sh
/mnt/c/Users/user1/Desktop/School/C++/matala1 a.out
Basiccheck.sh /mnt/c/Users/user1/Desktop/School/C++/matala1 a.out
Compilation      Memory Leak      Thread Race
PASS             PASS             FAIL
The returned value is : 1
tal@DESKTOP-LLPRHT1: /mnt/c/Users/user1/Desktop/School/C++/matala1$
```

טל נועם - 204397715  
אמיתי זמיר - 204324305