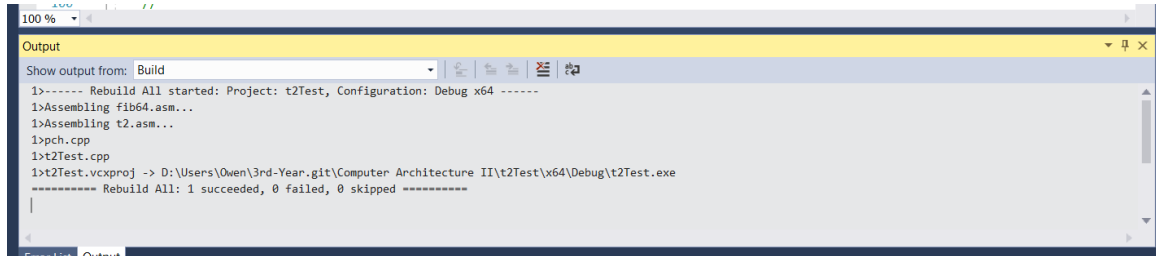


CS3021 Tutorial 2

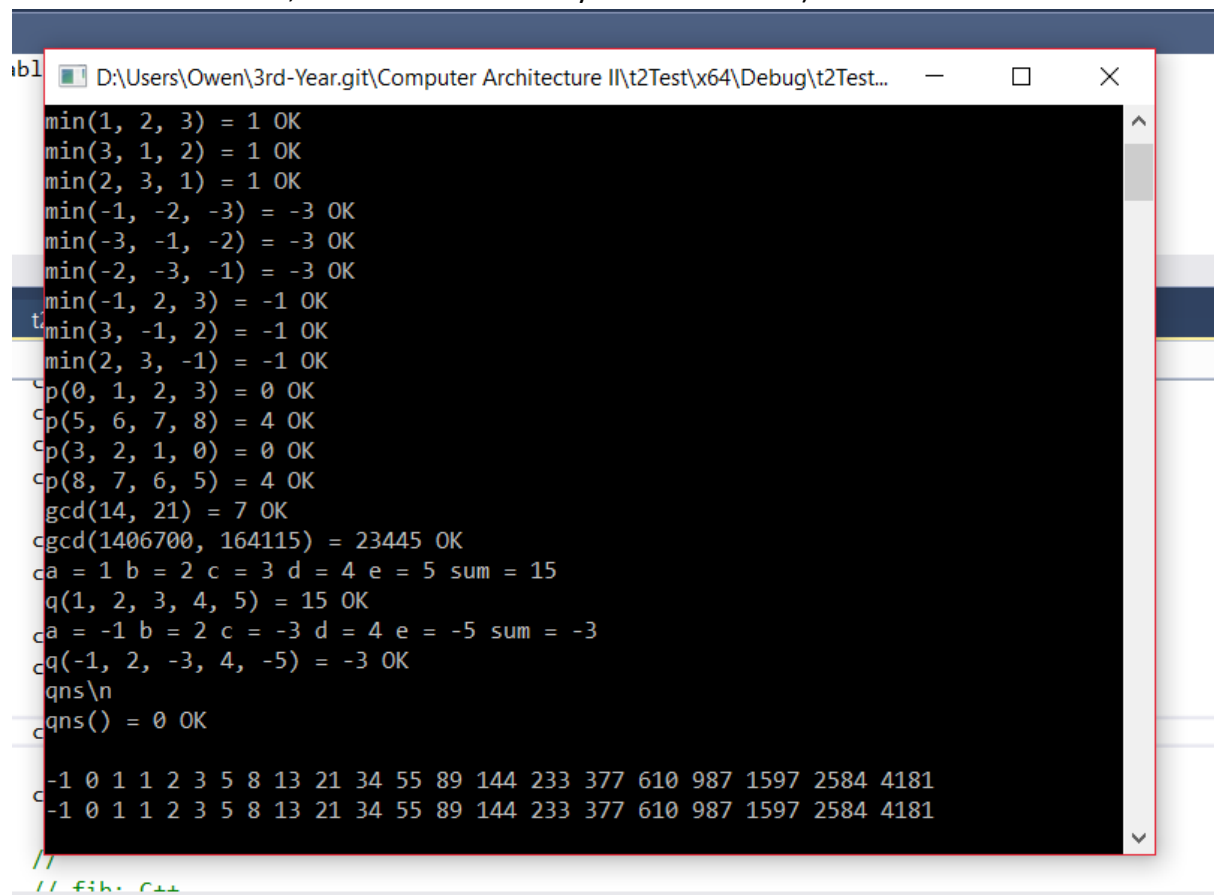
This is a statement that my program builds, runs and satisfies all tests. See the relevant screen shots. The files have also been included in the submission.

Build Screenshot:



```
100 %
Output
Show output from: Build
1>----- Rebuild All started: Project: t2Test, Configuration: Debug x64 -----
1>Assembling fib64.asm...
1>Assembling t2.asm...
1>pch.cpp
1>t2Test.cpp
1>t2Test.vcxproj -> D:\Users\Owen\3rd-Year.git\Computer Architecture II\t2Test\x64\Debug\t2Test.exe
***** Rebuild All: 1 succeeded, 0 failed, 0 skipped *****
```

Working Screenshot (note: The code for qns() that doesn't allocate shadow space is commented out here, as it causes a memory access violation):



```
D:\Users\Owen\3rd-Year.git\Computer Architecture II\t2Test\x64\Debug\t2Test...
min(1, 2, 3) = 1 OK
min(3, 1, 2) = 1 OK
min(2, 3, 1) = 1 OK
min(-1, -2, -3) = -3 OK
min(-3, -1, -2) = -3 OK
min(-2, -3, -1) = -3 OK
min(-1, 2, 3) = -1 OK
min(3, -1, 2) = -1 OK
min(2, 3, -1) = -1 OK
p(0, 1, 2, 3) = 0 OK
p(5, 6, 7, 8) = 4 OK
p(3, 2, 1, 0) = 0 OK
p(8, 7, 6, 5) = 4 OK
gcd(14, 21) = 7 OK
gcd(1406700, 164115) = 23445 OK
a = 1 b = 2 c = 3 d = 4 e = 5 sum = 15
q(1, 2, 3, 4, 5) = 15 OK
a = -1 b = 2 c = -3 d = 4 e = -5 sum = -3
q(-1, 2, -3, 4, -5) = -3 OK
qns\n
qns() = 0 OK

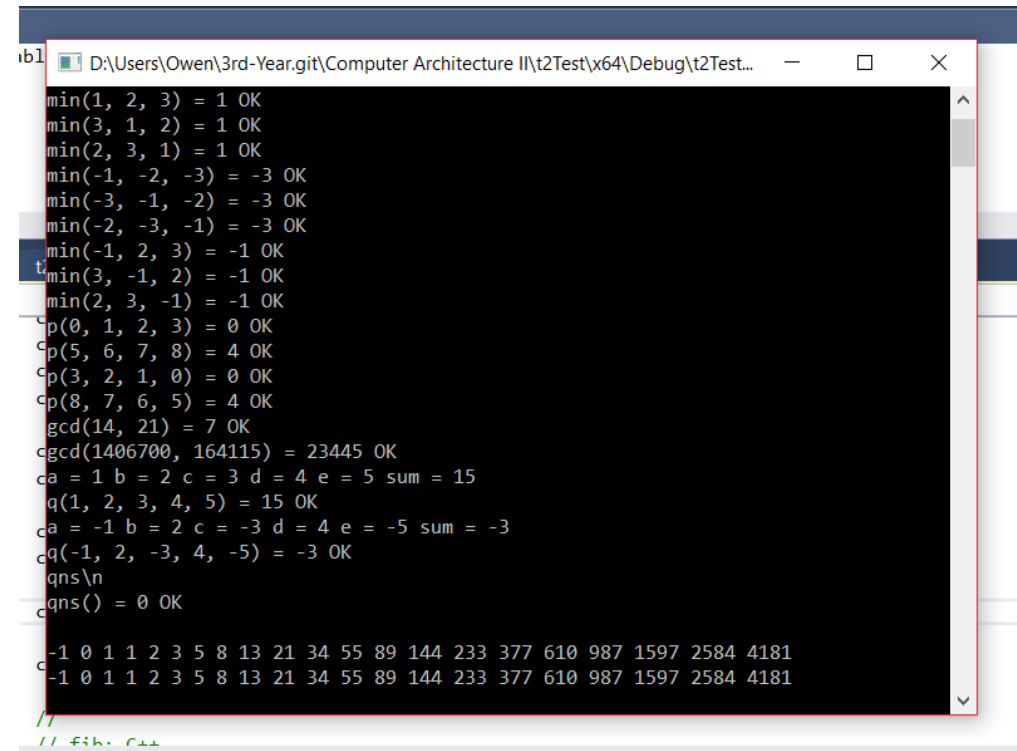
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
//
// fib.c++
```

Q2:

| |
|----------------|
| Shadow |
| Shadow |
| Shadow |
| Shadow |
| Return Address |
| Saved rbp |
| Shadow |
| Shadow |
| Shadow |
| Shadow |
| Return Address |
| Saved rbp |
| Shadow |
| Shadow |
| Shadow |
| Shadow |
| Return Address |
| Saved rbp |
| Shadow |
| Shadow |
| Shadow |
| Shadow |
| Return Address |
| Saved rbp |

Q4: When qns which calls printf() is executed with the allocation of shadow space, the function executes successfully without any errors. However, when the shadow space isn't allocated, a memory access violation occurs because printf() requires shadow space to run and if it is not provided the function attempts to access memory which doesn't "belong" to it. See the screenshots below:

With Shadow Space:



```
D:\Users\Owen\3rd-Year.git\Computer Architecture II\t2Test\x64\Debug\t2Test...
min(1, 2, 3) = 1 OK
min(3, 1, 2) = 1 OK
min(2, 3, 1) = 1 OK
min(-1, -2, -3) = -3 OK
min(-3, -1, -2) = -3 OK
min(-2, -3, -1) = -3 OK
min(-1, 2, 3) = -1 OK
min(3, -1, 2) = -1 OK
min(2, 3, -1) = -1 OK
p(0, 1, 2, 3) = 0 OK
p(5, 6, 7, 8) = 4 OK
p(3, 2, 1, 0) = 0 OK
p(8, 7, 6, 5) = 4 OK
gcd(14, 21) = 7 OK
gcd(1406700, 164115) = 23445 OK
ca = 1 b = 2 c = 3 d = 4 e = 5 sum = 15
q(1, 2, 3, 4, 5) = 15 OK
ca = -1 b = 2 c = -3 d = 4 e = -5 sum = -3
q(-1, 2, -3, 4, -5) = -3 OK
qns\n
qns() = 0 OK

-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
-1 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

//
// fib: C++
```

Without Shadow Space:

