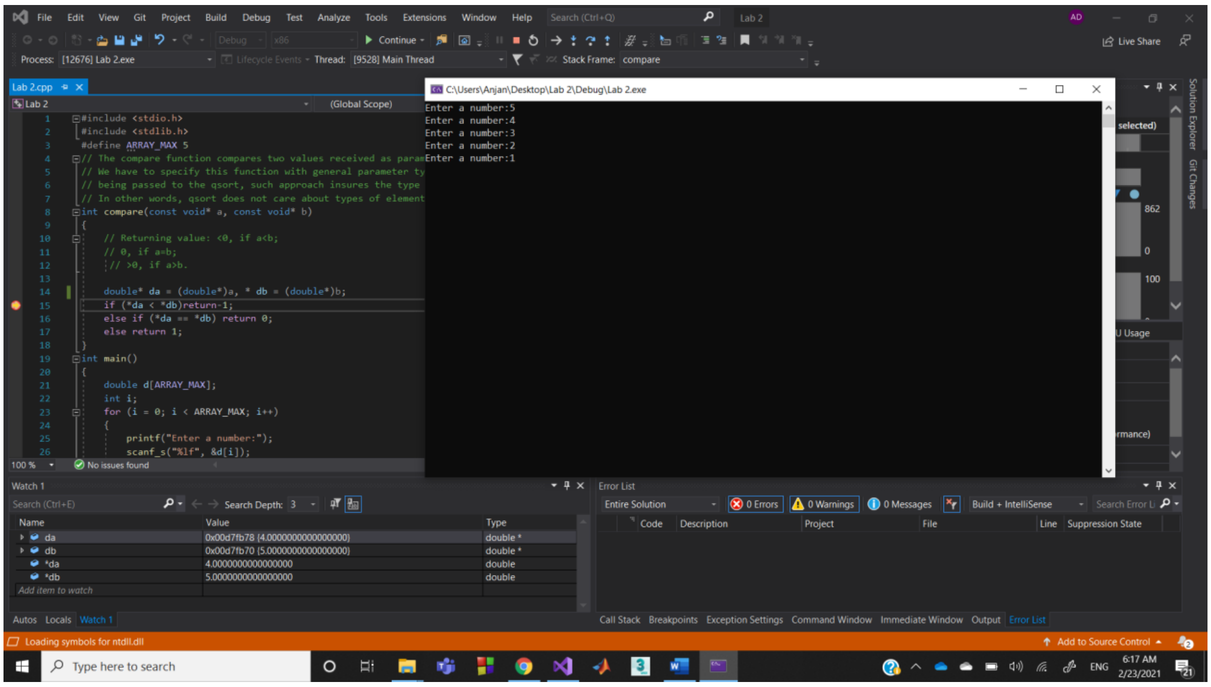
Task 1

Put a breakpoint inside the compare function, and check the values of pointer variables *da and *db. Is this code correct? Why do we use a function pointer? How can a function pointer be passed as a parameter? Why do the parameters of compare function have void* types? Please recall from C, why the last parameter of scanf should be a pointer.

Answer:



```
#include <stdio.h>
#include <stdlib.h>
#define ARRAY_MAX 5
// The compare function compares two values received as param
// We have to specify this function with general parameter ty
// being passed to the qsort, such approach insures the type
// In other words, qsort does not care about types of element
int compare(const void* a, const void* b)
{
    // Returning value: <0, if a<b;
    // 0, if a=b;
    // >0, if a>b.

    double* da = (double*)a, * db = (double*)b;
    if (*da < *db)return-1;
    else if (*da == *db) return 0;
    else return 1;
}
int main()
{
    double d[ARRAY_MAX];
    int i;
    for (i = 0; i < ARRAY_MAX; i++)
    {
        printf("Enter a number:");
        scanf_s("%lf", &d[i]);
```

```
Enter a number:5
Enter a number:4
Enter a number:3
Enter a number:2
Enter a number:1
```

Name	Value	Type
da	0x00d7fb78 {4.0000000000000000}	double *
db	0x00d7fb70 {5.0000000000000000}	double *
*da	4.0000000000000000	double
*db	5.0000000000000000	double

In the picture above we can see, that in Watch 1, the values of *da and *da are the first two input as the breakpoint is hit for the first time. So the code is working properly. Also their memory is different as shown, which are holding two values continuously and comparing them.

A function pointer is a variable that points to an address (starting) of an function , which can be used to call the function. It is useful because it sometime can help to change the behaviour of the function. Such as, in our sorting program some may want to sort the data in ascending order and others may want to do it in descending order. So if we allow the user to choose how to sort the data is simply to let the user pass in a function to the sort function. This function might take two pieces of data and perform a comparison on them.

To call the function pointed to by a function pointer, we treat the function pointer as though it were the name of the function we want to call.

The parameters of the compare function are void type, because we don’t know what type of variable will be passed on by the user. So it’s better to declare them as void and later typecasting to the type of variable we need.

The last parameter of scanf should be a pointer because we want to store the user input in an address. Also C only has pass-by-value parameters, so to pass a variable to put a value into, we have to pass its address.