C Programming Review

Question 1

Write a program that asks the user to enter two integers. The program should then print a message indicating whether the two integers are both positive or not. Zero is considered a positive integer. Running your program should for example look like this (where **3** and **0** are inputs typed by the user):

```
Enter the first integer: 3
Enter the second integer: 0
The integers 3 and 0 are both positive
```

or like this (where **-3** and **2** are inputs typed by the user):

```
Enter the first integer: -3
Enter the second integer: 2
The integers -3 and 2 are not both positive
```

Question 2

Write a program that asks the user to enter a floating point number and then prints the absolute value of the number. Your program must define and use a function called "**abs**" that takes a floating point number as argument and returns as result the absolute value of its argument. Printing the result must be done in the "**main**" function. Running the program should for example look like this (where **7.0** is an input typed by the user):

```
Enter a floating-point number: 7.0
The absolute value of 7.000000 is 7.000000
```

or like this (where **-3.5** is an input typed by the user):

```
Enter a floating-point number: -3.5
The absolute value of -3.500000 is 3.500000
```

Question 3

Write a program that asks the user to enter a start integer and an end integer, and then prints all the integers between these start and end integers. You can assume that the start integer is always less than the end integer (there is no need to check that in your program). Running the program should for example look like this (where **-3** and **19** are inputs typed by the user):

```
Enter the start integer: -3
Enter the end integer: 19
-3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

Question 4

Write a program that computes the product of many floating point numbers, where the number of floating point numbers is an integer specified by the user. Running the program should for example look like this (where **3**, **10.0**, **2.0**, and **3.0** are inputs typed by the user):

```
How many floating point numbers do you want to multiply together: 3
Enter a floating point number: 10.0
Enter a floating point number: 2.0
Enter a floating point number: 3.0
The product is 60.000000
```

or like this (where **2**, **3.0**, and **5.5** are inputs typed by the user):

```
How many floating point numbers do you want to multiply together: 2
Enter a floating point number: 3.0
Enter a floating point number: 5.5
The product is 16.500000
```

Question 5

Write a program that always reads exactly five floating point numbers from the user and puts them in an array. The program must define and use a function called "`array_abs`" that then takes the array as argument and changes each element of the array into the absolute value of the element. After calling the function, the "`main`" function should then print the content of the array back to the screen. Use loops and macros as much as possible in your program. Running your program should for example look like this (where `-2.0`, `5.0`, `-3.0`, `0.25`, and `-2.5` are inputs typed by the user):

```
Enter a floating point number: -2.0
Enter a floating point number: 5.0
Enter a floating point number: -3.0
Enter a floating point number: 0.25
Enter a floating point number: -2.5
The result is: 2.000000 5.000000 3.000000 0.250000 2.500000
```

Question 6

Write a program that reads from the user the month of the year and the day of the month and then prints the day number in the year. Your program must ask the user to enter again any input which is incorrect. Your program must define and use a macro `MONTHS` which is `12`, and must define and use the following array:

```
int days_in_month[MONTHS] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

Remember that array indexes start at `0` but days and months start at `1`. Running the program should for example look like this (where `6` and `19` are inputs typed by the user):

```
Enter the month: 6
Enter the day: 19
19/6 is the day number 170 in the year
```

Here is another example with some incorrect inputs (where `0`, `13`, `2`, `0`, `29`, and `28` are inputs typed by the user):

```
Enter the month: 0
Enter the month: 13
Enter the month: 2
Enter the day: 0
Enter the day: 29
Enter the day: 28
28/2 is the day number 59 in the year
```