Assignment 1 - The C Preprocessor and Bit Operations

- The problems of this assignment must be solved in C.

- Your programs should have the input and output formatting according to the testcases listed after the problems.

- Your programs should consider the grading rules:
  https://grader.eecs.jacobs-university.de/courses/320112/2019_1gA/Grading-Criteria-C2.pdf

Problem 1.1 *Swapping two variables*   (2 points)
Presence assignment, due by 18:30 h today   **Graded automatically with testcases only**
Write a macro and a program for swapping the contents of two variables. The macro should have three parameters: the two variables and the corresponding data type.
Your program should read two integers and two floats from the standard input. Then you should print on the standard output the contents of the four variables after swapping (floats with a floating point precision of 6).
You can assume that the input will be valid. Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 1.1: input

```
1
2
3.45
5.677
```

Testcase 1.1: output

```
After swapping:
2
1
5.677000
3.450000
```

Problem 1.2 *Determine the least significant bit*   (1 point)
Presence assignment, due by 18:30 h today   **Graded manually**
Write a macro and a program for determining the least significant bit (the first bit from the right in the binary representation) of an `unsigned char` read from the standard input.
Your program should read an `unsigned char` from the standard input and print the decimal representation of the `unsigned char` as well as its least significant bit (which is either 1 or 0) on the standard output using only bitwise operators and without explicitly converting to binary representation.
You can assume that the input will be valid. To pass the testcases your output has to be identical with the provided ones.

Testcase 1.2: input

```
F
```

Testcase 1.2: output

```
The decimal representation is: 70
The least significant bit is: 0
```

Problem 1.3 *Determine the mid-range of three values*   (2 points)
Due by Tuesday, February 19th, 10:00 AM   **Graded automatically with testcases only**
Write multiple macros and a program for determining the mid-range of three values. The mid-range of three variables `a`, `b`, and `c` is calculated as

$$mid\_range(a, b, c) = \frac{\min(a, b, c) + \max(a, b, c)}{2}.$$

For example if 3, 10, 1 is the input, the mid-range of these values is

$$mid\_range(3, 10, 1) = \frac{\min(3, 10, 1) + \max(3, 10, 1)}{2} = \frac{1 + 10}{2} = \frac{11}{2} = 5.5.$$

Your program should read three integers from the standard input. For calculating the mid-range of these values only macros should be used. The mid-range should be printed on the standard output with a floating point precision of 6.

You can assume that the input will be valid. Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 1.3: input

```
3
10
1
```

Testcase 1.3: output

```
The mid-range is: 5.500000
```

Problem 1.4 *Conditional compilation for showing intermediate results*   (2 points)
Due by Tuesday, February 19th, 10:00 AM   **Graded manually**
Write a program which computes the scalar product of two n-dimensional integer vectors and uses conditional compilation for showing/not showing intermediate results (products of the corresponding components). The scalar product of two n-dimensional vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ is calculated as

$$< x, y >= \sum_{i=1}^{n} x_i \cdot y_i.$$

For example the scalar product of the vector $x = (1, 2, 3)$ with the vector $y = (3, 5, 1)$ is

$$< x, y >= 1 \cdot 3 + 2 \cdot 5 + 3 \cdot 1 = 3 + 10 + 3 = 16.$$

The intermediate results which are to be shown or not are 3, 10 and 3.

Your program should read from the standard input the dimension of the vector (in the previous example 3) along with the components of two integer vectors. The output consists of the intermediate results and the value of the scalar product of the two vector if the directive INTERMEDIATE is defined. If INTERMEDIATE is not defined then only the scalar product of the two vectors should be printed on the standard output.

You can assume that the input will be valid. To pass the testcases your output has to be identical with the provided ones.

Testcase 1.4: input

```
3
1
2
3
3
5
1
```

Testcase 1.4: output

```
The intermediate product values are:
3
10
3
The scalar product is: 16
```

Problem 1.5 *Binary representation backwards*   (1 point)
Due by Tuesday, February 19th, 10:00 AM   **Graded automatically with testcases only**
Write a program using bit masks and bitwise operators for printing the binary representation of an `unsigned char` backwards. For example the character '2' is encoded as 50 in decimal representation which is in binary representation 110010. Therefore, the backwards binary representation is 010011.

Your program should read an `unsigned char` from the standard input and print on the standard output the backwards binary representation of the read character without explicitly converting the decimal value to binary or using an array to store the bits.

You can assume that the input will be valid. Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 1.5: input	**Testcase 1.5: output**
2	The decimal representation is: 50 The backwards binary representation is: 010011

Problem 1.6 *Binary representation*   (2 points)

Due by Tuesday, February 19ᵗʰ, 10:00 AM   **Graded manually**

Write a program using bit masks and bitwise operators for printing the binary representation of
an `unsigned char` without storing the bits in an array or explicitly converting to binary. For
example the character '2' is encoded as 50 in decimal representation which is in binary represen-
tation on 8 bits 00110010.

Your program should read an `unsigned char` from the standard input and print on the stan-
dard output the binary representation of the read character.

*You can assume that the input will be valid. To pass the testcases your output has to be identical
with the provided ones.*

Testcase 1.6: input	**Testcase 1.6: output**
2	The decimal representation is: 50 The binary representation is: 00110010

Problem 1.7 `set3bits()`   (2 points)

Due by Tuesday, February 19ᵗʰ, 10:00 AM   **Graded automatically with testcases only**

Write a program for setting three bits of an `unsigned char` to 1. The function `set3bits` should
have four parameters: the `unsigned char` to be changed and the three bits which are to be set
to 1. For example the character '2' is encoded as 50 in decimal representation which is in binary
representation on 8 bits 00110010. If `set3bits()` with bits 7, 6 and 1 to be set to 1 is called then
the output on the standard output should be 11110010. Print the result on the standard output
from the `main()` function.

*You can assume that the input will be valid. Your solution has to satisfy the requirements from
the problem description and has to pass the following testcase and potentially other testcases
which are uploaded. All characters are relevant for passing testcases including newlines and
spaces.*

Testcase 1.7: input	**Testcase 1.7: output**
2 7 6 1	The decimal representation is: 50 The binary representation is: 00110010 After setting the bits: 11110010

How to submit your solutions

- Your source code should be properly indented and compile with `gcc` without any warnings (You
  can use `gcc -Wall -o program program.c`). Insert suitable comments (not on every line . . .) to
  explain what your program does.

- Please name the programs according to the suggested filenames (they should match the description
  of the problem) in Grader. Otherwise you might have problems with the inclusion of header files.
  Each program **must** include a comment on the top like the following:
  ```
  /*
     JTSK-320112
     a1_p1.c
     Firstname Lastname
     myemail@jacobs-university.de
  */
  ```

- You have to submit your solutions via *Grader* at
  https://cantaloupe.eecs.jacobs-university.de.
  If there are problems (but **only** then) you can submit the programs by sending mail to
  `k.lipskoch@jacobs-university.de` **with a subject line that begins with JTSK-320112.**
  **It is important that you do begin your subject with the coursenumber, otherwise I might have
  problems to identify your submission.**

- Please note, that after the deadline it will not be possible to submit any solutions. It is useless to send
  late solutions by mail, because they will not be accepted.

This assignment is due by Tuesday, February 19ᵗʰ, 10:00 AM.