Programming in C II, Group A
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: JTSK-320112
February 8th, 2018

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/320112/2018_1gA/Grading-Criteria-C2.pdf

Problem 1.1 *Circular permutation of three variables* (2 points)

Presence assignment, due by 18:30 h today

Write a macro and a program for the circular permutation of the contents of three variables having the same data type (i.e., the content of the first is put into the second, the content of the second into the third and the content of the third into the first). The macro should have four parameters: the three variables and their corresponding data type.
Your program should read three integers and three doubles from the standard input. You should print on the standard output the contents of the six variables after the permutation (doubles with 5 after floating point precision).
You can assume that the input will be valid.

Testcase 1.1: input

```
1
2
3
3.45
5.677
8.98273
```

Testcase 1.1: output

```
After the permutation:
3
1
2
8.98273
3.45000
5.67700
```

Problem 1.2 *Determine the third least significant bit* (1 point)

Presence assignment, due by 18:30 h today

Write a macro and a program for determining the third least significant bit (the third 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 third 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.

Testcase 1.2: input

```
F
```

Testcase 1.2: output

```
The decimal representation is: 70
The third least significant bit is: 1
```

Problem 1.3 *Determine the value of an expression* (2 points)

Write multiple macros and a program for determining the value of the following expression depending on three variables `a`, `b`, and `c` calculated as

$$\texttt{expr}(\texttt{a},\texttt{b},\texttt{c}) = \frac{\texttt{sum}(\texttt{a},\texttt{b},\texttt{c}) + \texttt{max}(\texttt{a},\texttt{b},\texttt{c})}{\texttt{min}(\texttt{a},\texttt{b},\texttt{c})}.$$

For example if 3, 10, 2 is the input, the value of the expression is

$$\texttt{expr}(3,10,2) = \frac{\texttt{sum}(3,10,2) + \texttt{max}(3,10,2)}{\texttt{min}(3,10,2)} = \frac{15+10}{2} = \frac{25}{2} = 12.5.$$

Your program should read three integers from the standard input. For calculating the expression for these values only macros should be used. The result should be printed on the standard output with a floating point precision of 6.
You can assume that the input will be valid.

Testcase 1.3: input

```
3
10
2
```

Testcase 1.3: output

```
The value of the expression is: 12.500000
```

Problem 1.4 *Conditional compilation for showing intermediate results* (2 points)

Write a program which computes the product of two $\texttt{n} \times \texttt{n}$ integer matrices and uses conditional compilation for showing/not showing intermediate results (products of the corresponding components). The product of two $\texttt{n} \times \texttt{n}$ matrices $A = (A_{\texttt{ij}})$ and $B = (B_{\texttt{jk}})$ with $\texttt{i}, \texttt{j}, \texttt{k} = 1, \ldots \texttt{n}$ is calculated as

$$C_{\texttt{ik}} = \sum_{\texttt{j}=1}^{\texttt{n}} A_{\texttt{ij}} \cdot B_{\texttt{jk}}, \text{ with } \texttt{i} = 1, \ldots, \texttt{n} \text{ and } \texttt{k} = 1, \ldots, \texttt{n}.$$

For example the product of the matrices $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ is

$$C = \begin{pmatrix} 1\cdot 1 + 2\cdot 3 & 1\cdot 2 + 2\cdot 4 \\ 3\cdot 1 + 4\cdot 3 & 3\cdot 2 + 4\cdot 4 \end{pmatrix} = \begin{pmatrix} 1+6 & 2+8 \\ 3+12 & 6+16 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}.$$

The intermediate results which are to be shown or not are 1, 6, 2, 8, 3, 12, 6, and 16.
Your program should read from the standard input the dimension of the matrix (in the previous example 2) along with the components of two integer matrices. The output consists of the intermediate results and the value of the product of the two matrices if the directive INTERMEDIATE is defined. If INTERMEDIATE is not defined then only the product of the two matrices should be printed on the standard output.
You can assume that the input will be valid.

Testcase 1.4: input

```
2
1
2
3
4
1
2
3
4
```

Testcase 1.4: output

```
The intermediate product values are:
1
6
2
8
3
12
6
16
The product of the matrices is:
7 10
15 22
```

Problem 1.5 *Binary representation backwards* (1 point)

Write a program using bit masks and bitwise operators for printing the binary representation of an unsigned int backwards. For example the binary representation of the unsigned int 12345 on 16 bits is 0011000000111001. Therefore, the backwards binary representation is 10011100000011.
Your program should read an unsigned int from the standard input and print on the standard output the backwards binary representation of the read integer without leading zeros. You should not store the bits in an array.
You can assume that the input will be valid.

Testcase 1.5: input

```
12345
```

Testcase 1.5: output

```
The backwards binary representation is: 10011100000011
```

Problem 1.6 *Binary representation* (2 points)

Write a program using bit masks and bitwise operators for printing the binary representation of an unsigned int on 16 bits without storing the bits in an array. For example the binary representation of the unsigned integer number 12345 on 16 bits is 0011000000111001.
Your program should read an unsigned int from the standard input and print on the standard output the binary representation of the integer number with leading zeros.
You can assume that the input will be valid.

Testcase 1.6: input

```
12345
```

Testcase 1.6: output

```
The binary representation is: 0011000000111001
```

Problem 1.7 `setswitchbits()` (2 points)

Write a program for setting one bit to 1 and switching another bit of an unsigned int. The function `setswitchbits` should have three parameters: the unsigned int to be changed and the two bits. The first bit is to be set to 1 and the other bit is to be switched. For example the binary representation on 16 bits of the unsigned int 12345 is 0011000000111001. If `setswitchbits()` with bits 6 (the 7th bit from the right) and 12 (the 13th bit from the right) is called then the output on the standard output should be 0010000001111001.
You can assume that the input will be valid.

Testcase 1.7: input

```
12345
6
12
```

Testcase 1.7: output

```
The binary representation is: 0011000000111001
After setting and switching: 0010000001111001
```

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 13th, 10:00 h.