

CSC / CIS 175

Problem Solving and Programming - I

University of Michigan-Flint
Department of Computer Science, Engineering, and Physics (CSEP)



COLLEGE OF ARTS & SCIENCES

COMPUTER SCIENCE, ENGINEERING, & PHYSICS

Fall 2013

October 20, 2013

Homework 6

(100 points)

due by October 30, Wednesday 8:00am

Remarks:

- No emailed homeworks will be accepted.
 - Only submission is via the BB system.
 - No late submissions will be accepted.
 - Individual assignment. No collaboration or group work is
-

Questions for the deliverable:

1. A company wants to transmit data over the Internet but is concerned about the security of the sensitive data. All of the data are transmitted as four-digit integers. The company asked you to write a program that encrypts the data so that it can be transmitted securely. Your program should read a four-digit integer and encrypt it as follows: Replace each digit (the sum of that digit plus 7) modulus 10. Then, swap the first digit with the third, swap the second with the fourth and print the encrypted integer. Add decryption functionality to your program above that inputs an encrypted four-digit integer and decrypts it to form the original number.

Expected Output:

(i)

Enter a four-digit number: 3456

Encrypted number is 2301

Enter an encrypted number: 2301
Decrypted number is 3456

(ii)

Enter a four-digit number: 0005
Encrypted number is 7277
Enter an encrypted number: 7277
Decrypted number is 5

2. Write a program that prints a table of binary, octal and hexadecimal equivalents of the decimal numbers in the range 0 through 255.

Expected Output:

Decimal	Binary	Octal	Hex
=====			
0	00000000	000	00
1	00000001	001	01
2	00000010	002	02
3	00000011	003	03
4	00000100	004	04
5	00000101	005	05
6	00000110	006	06
7	00000111	007	07
8	00001000	010	08
9	00001001	011	09
10	00001010	012	0a
11	00001011	013	0b
12	00001100	014	0c
13	00001101	015	0d
14	00001110	016	0e
15	00001111	017	0f
16	00010000	020	10
17	00010001	021	11
18	00010010	022	12
19	00010011	023	13
20	00010100	024	14
21	00010101	025	15
22	00010110	026	16
23	00010111	027	17
24	00011000	030	18
25	00011001	031	19
26	00011010	032	1a
27	00011011	033	1b
28	00011100	034	1c

29	00011101	035	1d
30	00011110	036	1e
31	00011111	037	1f
32	00100000	040	20
33	00100001	041	21
34	00100010	042	22
...			
...			
...			
241	11110001	361	f1
242	11110010	362	f2
243	11110011	363	f3
244	11110100	364	f4
245	11110101	365	f5
246	11110110	366	f6
247	11110111	367	f7
248	11111000	370	f8
249	11111001	371	f9
250	11111010	372	fa
251	11111011	373	fb
252	11111100	374	fc
253	11111101	375	fd
254	11111110	376	fe
255	11111111	377	ff

3. Write a program that reads five numbers (between 1 and 30). Assume that the user only enters valid values. For each number that is read, your program should output a line containing that number of adjacent asterisks.

Sample Output:

```
Enter 5 numbers between 1 and 30: 23
*****
2
**
30
*****
11
*****
5
*****
```

4. Write a program that uses for statements to print the following separately, one below the other. Use for loops to generate the patterns. All asterisks should be printed by a single statement of the form (`cout << '*'`; and `cout << " "`).

Expected Output:

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

5. Write a simple calculator program that adds, subtracts, multiplies and divides. When the program is run, it initializes the result to 0. The user can then type in an operator and number. The result is updated and displayed. The following operators are valid:

Operator	Meaning
-----	-----
+	Addition
-	Subtraction
*	Multiplication
/	Division
-----	-----

Handle the case when the user enters q or Q to quit and h or H for usage help.

Sample Output:

Result: 0

Enter operator from the table below, Q or q for quit, H or h for help :

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply
/	Divide

+

Enter the value : 23

0 + 23

Result: 23

Enter operator from the table below, Q or q for quit, H or h for help :

/

Enter the value : 0

Error:Divide by zero. Operation ignored

Result: 23

Enter operator from the table below, Q or q for quit, H or h for help :

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply
/	Divide

*

Enter the value : 3

23 * 3

Result: 69

Enter operator from the table below, Q or q for quit, H or h for help :

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply
/	Divide

-

Enter the value : 56

69 - 56

Result: 13

Enter operator from the table below, Q or q for quit, H or h for help :

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply
/	Divide

/

Enter the value : 4

13 / 4

Result: 3.25

Enter operator from the table below, Q or q for quit, H or h for help :

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply
/	Divide

u

Invalid Entry! Please retry.

Enter operator from the table below, Q or q for quit, H or h for help :

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply

```

/      Divide
h
Use '+' to add a value to the result ( 3.25 ),
Use '-' to subtract a value from the result ( 3.25 ),
Use '*' to multiply a value with the result ( 3.25 ),
Use '/' to divide a value into the result ( 3.25 ),
Result: 3.25
Enter operator from the table below, Q or q for quit, H or h for help :

```

Operator	Meaning
=====	=====
+	Add
-	Subtract
*	Multiply
/	Divide

```

q
Thank you for using our calculator. Come back again :-)

```

6. Write a function `IntegerPower(base, exponent)` that returns the value of $base^{exponent}$. For example, $IntegerPower(3,4) = 3^4 = 3 * 3 * 3 * 3 = 81$. Assume that exponent is positive, nonzero integer, and base is an integer. The function should use for or while loop to control the calculation. Do not use any math library functions.

Expected Output:

```

(i)
Enter the base : 3
Enter the exponent : 4
3 to the power of 4 is 81.000000

(ii)
Enter the base : 2
Enter the exponent : 24
2 to the power of 24 is 16777216.000000

```

7. **Extra credit (7%)**: Combine your code for q4 above to print the triangles side by side as shown below:

Expected Output:

```

*               *
**              *
***             *
****            *
*****           *
*****          *
*****         *
*****        *
*****       *
*****      *
*****     *
*****    *
*****   *
*****  *
***** *
*****

```

Deliverables:

1. Source Code: (.cpp file) that must start with a comment block similar to the following:

```

/*****
** Author          : Suleyman Uludag
** Program         : hw1, q1
** Date Created    : September 15, 2013
** Date Last Modified : September 16, 2013
** Usage          : No command line arguments
**
** Problem:
Accept the following information from the user (keyboard):
- Hw1, hw2 and hw3 (out of 100)
- Midterm (out of 100)
- Final exam (out of 100)
Calculate the total grade out of 100 based on the following grading scale:
Hws          -->    30% (10% each)
Midterm      -->    30%
Final Exam   -->    40%
** Pseudocode:
** 1)
** 2)
*****/
```

2. Executable (.exe file under windows). You must explicitly state the platform of your executable (such as Linux, etc.) if it is not Windows. Please name your file by using the question number: **hw1-q1.exe** (for Windows)
3. Screenshot of your app. For screenshot, you can use the following free program on windows:

<http://www.wisdom-soft.com/downloads/setupsscreenhunterfree.exe>

For Linux/Unix, there are many alternatives. I personally like shutter.

File naming convention example:

hw1-q1.png (or .jpg or another graphics format)

4. You must zip all the above three files into ONE .zip file and submit your assignment by the deadline on moodle system. Name your file as Lastname-Firstname-hw#.zip. For example, **Uludag-Suleyman-hw1.zip**

For generating .zip file, you may use the following free software on Windows:

<http://www.7-zip.org/download.html>

Linux/Unix has many built-in.
