

Ascii Encryption

Your Tasks (Mark these off as you go)

- ☐ Interpret a *while* loop
- ☐ Encrypt and decrypt a word using the letters' corresponding ascii values
- ☐ Determine the ascii equivalent of a char and vice versa
- ☐ Write a *while* loop the encrypts/decrypts a word provided by a user
- ☐ Write an algorithm the encrypts/decrypts a word using the Caesar Cipher
- ☐ Receive credit for this lab guide

☐ Interpret a *while* loop

As we learned previously, one of the most important control structures in Java is the *while* loop. A *while* loop is a block of code that is repeated until a condition is no longer true.

Consider a loop that sums all the numbers in a given range. To do this we need a starting point and an ending point. We also need to know how to increment each step along the way.

- *start* = 0
- *incrementing* = +1
- *stop* = 100

The problem of summing all the number in a given range can be solved with the following loop.

```
int sum = 0;
int stop = 100;
int start = 0;

while(start < stop){

    sum += start;
    start++;
}

System.out.println(sum);
```

Write a loop that prints all the even numbers in a given range.

When writing loops keep in mind the following

1. Initializing expression. The initializing expression can be any integer.
2. Control expression. The control expression indicates how long to continue looping. This is a Boolean expression. As long as the expression is true, the loop will continue.

Warning: There is something really bad that can happen here. You must write your code so as to ensure that this control statement will eventually become false, thus causing the loop to terminate. Otherwise you will have an endless loop which will crash your program.

3. Step expression. The step expression tells us how our variable should change each time through the loop. In this case we are incrementing j each time. However, other possibilities could include,

n -- n = n + 4 n = n * 3 etc.

Indicate the output for each of the following,	
<pre>int j = 0, g = 1; while(g < 10){ j++; g++; } System.out.println(j);</pre>	
<pre>int s = 1, j = 4; while(j >= 0){ s = s + j; j--; } System.out.println(s);</pre>	
<pre>int j = 0; i = 10; while(i > 0){ i *= -3; i--; } System.out.println(i)</pre>	

❑ Encrypt and decrypt a word using the letters' corresponding ascii values

Now that we understand a bit about *while* loops, let's consider how a *while* loop can be applied to perform some simple decryption.

First let's consider the alphabet and the corresponding ascii value of each letter,

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90

Using only the indices of each letter, a user could encrypt the word DOG as follows,

687971

Where 68 = D, 79 = O, and 71 = G

Notice that in our encrypted message two digits represent a single letter.

For each of the encrypted words, indicate their value, 8773846772 7172798384
Encrypt the words below, GOBLIN VAMPIRE

❑ Determine the ascii equivalent of a char and vice versa

The char data type is a primitive data type used to represent symbols (alphabets, numbers, etc). The size of the a Java char is 16 bits and ranges between 0 and 65,536.

A char in Java can be created using the char key word,

```
char c;
```

Symbols and letters can be used to initialize a char variable by enclosing the value in single quotes,

```
char letterA = 'A';
```

The ascii equivalent or symbols and letters can also be used to initialize char variable. This is illustrated below,

Code	Output
<pre>char letterA = 65; char letterB = 'B'; char letterC = 67; System.out.println("The characters are " + letterA + letterB + letterC);</pre>	The characters are ABC

Its also possible to determine the numerical value of a char by simply assigning a symbol to an `int`. The below code is allowed because while a `char` is only 16 bits, an `int` is 32 bits. Assigning a `char` to an `int` does not result in data loss.

Code	Output
<pre>int asciiJ = 'J'; System.out.println(asciiJ);</pre>	74

The reverse however is not allowed,

Code	Output
<pre>int asciiJ = 74; char j = asciiJ;</pre>	incompatible types: possible lossy conversion from int to char char j = asciiJ;

Assigning an `int` to a `char` requires that the `int` be casted. The below code is useful because it allows us to identify the symbolic equivalent of an `int` as long as it is within the range of `char` values.

Code	Output
<pre>int asciiJ = 74; char j = (char)asciiJ; System.out.println(j);</pre>	J

Indicate what is printed. If an error occurs, indicate how to fix the error.	
<pre>char letter = 70; int num = letter; System.out.println(num);</pre>	
<pre>char letter = 'K'; int num = letter; System.out.println(num);</pre>	
<pre>int num = 656667; char letter = num%100; System.out.println(letter);</pre>	
<pre>char secretLetter = 71; char temp = secretLetter; System.out.println(temp);</pre>	

□ Write a *while* loop that decrypts/encrypes a word provided by the user

Notice that all the capital letters of the alphabet can be decoded in two digits: 65 thru 97. Also, recall that the last digits of any number can be retrieved using modulus. For example,

Code	Output
<pre>int num = 656667; System.out.println(num%100); num /= 100; System.out.println(num%100); num/=100; System.out.println(num%100);</pre>	67 66 65

Write a while loop that could be used to obtain all the ascii equivalents for letters that make up the secret message below,

```
int num = 8773846772;
```

How could you modify the loop to determine the symbol that corresponds to each number?

The charAt method is useful for retrieving the char at a specific index in a word. Consider the following example,

Code	Output
<pre>String word = "VAMPIRE"; int letter = 0; while(letter < word.length()){ char l = word.charAt(letter); System.out.println(l); letter++; }</pre>	V A M P I R E

Modify the loop above to print out the ascii equivalent of each character of word.

□ Write an algorithm that encrypts/decrypts a word using a Caesar Cipher and ascii

The Caesar Cipher is an encryption method that replaces each letter in a text by a letter some fixed number of positions down the alphabet. For example, the word CODE could be encrypted with a shift of 2 as EQFG. Below are some more examples,

Word	Shift	Encrypted
CODE	4	GSHI
TIMBERLINE	17	KZDSVICZEV
JAVA	12	ARMR

Below are some encrypted words. Can you figure out the shift and the original word?		
Encrypted	Shift	Word
WIZURP		
IAXHQE		

Consider the word defined below and the corresponding shift. Write an algorithm that could be used to decrypt the word using the Caesar Cipher. Your algorithm should work for any word and any shift.

```
String word = "VAMPIRE";  
int shift = 7;
```

Consider the encrypted word below and the corresponding shift. Write an algorithm that could be used to decrypt the word using the Caesar Cipher. Your algorithm should work for any word and any shift.

```
String word = "LIHHDR";  
int shift = -7;
```

☐ **Receive Credit for this lab guide**

Submit this portion of the lab to Pluska to receive credit for the lab guide. Once received, your completed code challenges will also be graded and will count towards your final lab grade.