



PRIFYSGOL
BANGOR
UNIVERSITY

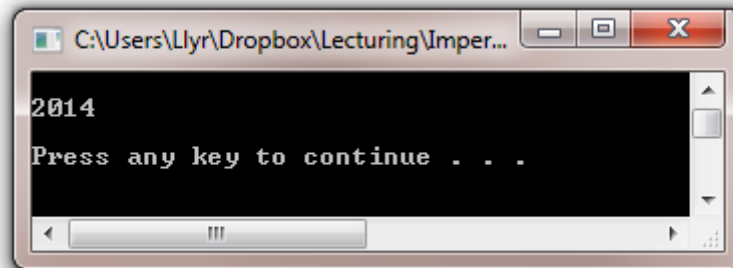
Imperative Programming in C

Laboratory 4: Iteration and Arrays

Dr. Llyr ap Cenydd

Exercise 1 – No Numbers Allowed

Your task for this exercise is to write a program that prints out the number 2014 on screen. The only catch is you are not allowed to use numbers in your source code. Your output should look like this:



Exercise 2 - Sequences

Write a program that generates the following sequences:

- 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
- -10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10
- 1, 2, 3, 4, 5, 4, 3, 2, 1
- The ASCII English alphabet (uppercase)
- The ASCII English alphabet (lowercase)
- English alphabet without vowels (a e i o u)

Hints

- For-loops are ideal for generating sequences like this
- Every ASCII character has a decimal value
- ASCII characters table: <http://www.asciitable.com/>
- If you need to reverse a sequence, try having a variable that can flip on and off to reverse the direction when triggered

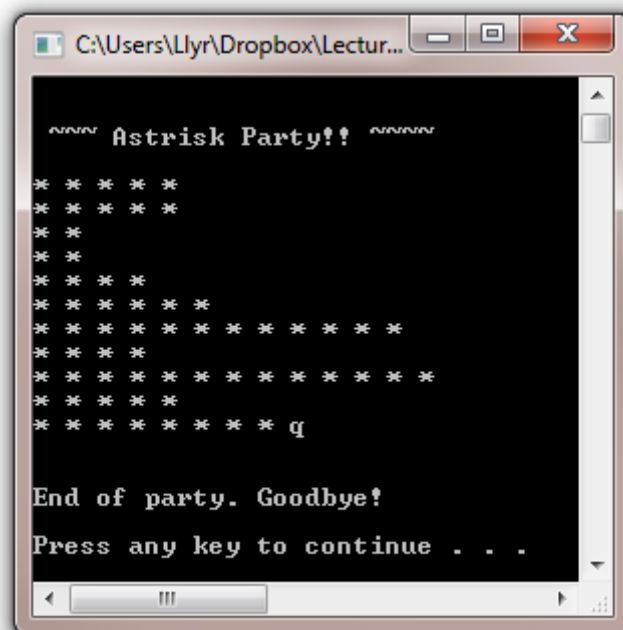
Exercise 3 – Asterisk Party

Your task for this exercise is to write a program that will print out a random number of asterisk (*) characters on screen every time the user hits the enter key. If the user types in the letter 'Q', the program should end.

Hints

- `getchar()` will listen for a single keypress and return the value
- You can use a sentinel controlled while loop to create a loop that will only end when the sentinel value is `q`
- Alternatively you can create an infinite while loop using `while(1)`, and use the `break` command to exit the loop

Example output:



Exercise 4 – Mock menu system

You have been asked to write a menu system for a video game. The menu choices are

1. Single Player
2. Multiplayer
3. Options
4. Credits
5. Quit

Use Switch case statements and a while loop to construct a main menu system. The menu system should follow this order:

1. Display main menu
2. Ask user to enter their choice
3. When user enters 1-4, print appropriate message, wait a few seconds
4. Clear screen
5. Goto 1

If the user chooses quit option, the program should exit.

Hints

- Consider using a do-while loop for this exercise
- You can clear the screen using the command **System("CLS")** in windows;
- You can use the command **sleep(value)** to ask the program to sleep for a certain amount of time in milliseconds (1000 = 1 sec).
- The function sleep can be found in [`<time.h>`](#)
- You can exit a while loop using the break keyword



Basic Arrays

There are times when we need to store a complete list of numbers or other data items. We could just create a variable to store every one of the items in a list, but this is inefficient – just to create five pieces of data we would need to do something like this:

```
int v1,v2,v3,v4,v5;
```

What happens if we want to store 1000 data items?

In C (and all modern programming languages), **arrays** allow us to store information **of the same data type** under a single variable name. To create an array called myArray of five integers in C we would simply write:

```
int myArray[5];
```

To print the value of the array at a particular **index**, we use the following notation:

```
printf( "%d" , myArray[2]);
```

which would print the 3rd item in the array (as index 0 is 1st item).

In order to modify an array, we can once again use the index:

```
myArray [3] = 100;
```

Which would store the value of 100 in index 3 of the array.

If we don't know the exact size (or "length") of an array in C, we can use the sizeof operator like this:

```
int arraySize = sizeof(myArray);
```

However this will actually return the size of the array in bytes. For example, if we are storing 5 integers, the size will usually be $5 * 4 \text{ bytes} = 20$.

If you want the actual number of items in an array, we have to divide this value by the size of one of the elements in the array, in this case the size of an integer, which is 4.

To go one step further we could even define a Macro at the top of our program that does this:

```
#define N_ELEMENTS(array) (sizeof(array)/sizeof((array)[0]))
```

Which we could then use to calculate the number of variables stored in an array:

```
int arraySize = N_ELEMENTS(myArray);
```

Strings

In C, there is no String data type. However, a string is just a 'string' of characters. Therefore, to create a string in C we simply create an array of chars like this:

```
char myString [7] = "Lemons" ; //create a string of size 7
```

We need to define the size to be one larger than the number of characters because there needs to be space in memory to store `"/0"` which tells the compiler where the string ends in memory.

We can also create a string without defining its size like this:

```
char myString [] = "Lemons, Limes and Coconuts" ;
```

If you create a character array and you don't know the number of letters stored in the array (for example you might want to loop through every character in the array), we can use the function `strlen()` from the `string.h` library like this:

```
#include <stdio.h>
```

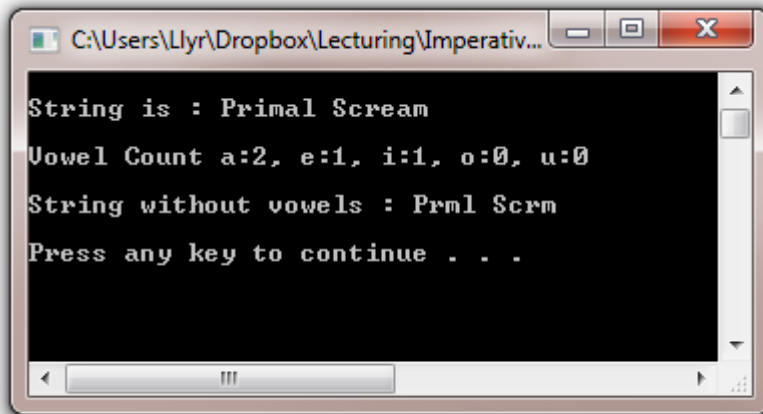
```
char szInput[256];  
printf ( "Enter a sentence: " );  
fgets (szInput, 256, stdin );  
printf ( "The sentence entered is %u characters long.\n"  
,(unsigned)strlen(szInput));
```

In this case, the `fgets()` function allows us to take a string from `stdin` (of size 256 here) or until the newline character or end-of-file is reached.

Exercise 5 – Vowel Exterminator

Write a program that asks the user to type in a string. The program should then count the vowels (a e i o u) in the string and print out the count of each letter. The program should also print out the string with all vowels removed.

Example output:



```
C:\Users\Llyr\Dropbox\Lecturing\Imperativ...
String is : Primal Scream
Vowel Count a:2, e:1, i:1, o:0, u:0
String without vowels : Prml Scrm
Press any key to continue . . .
```

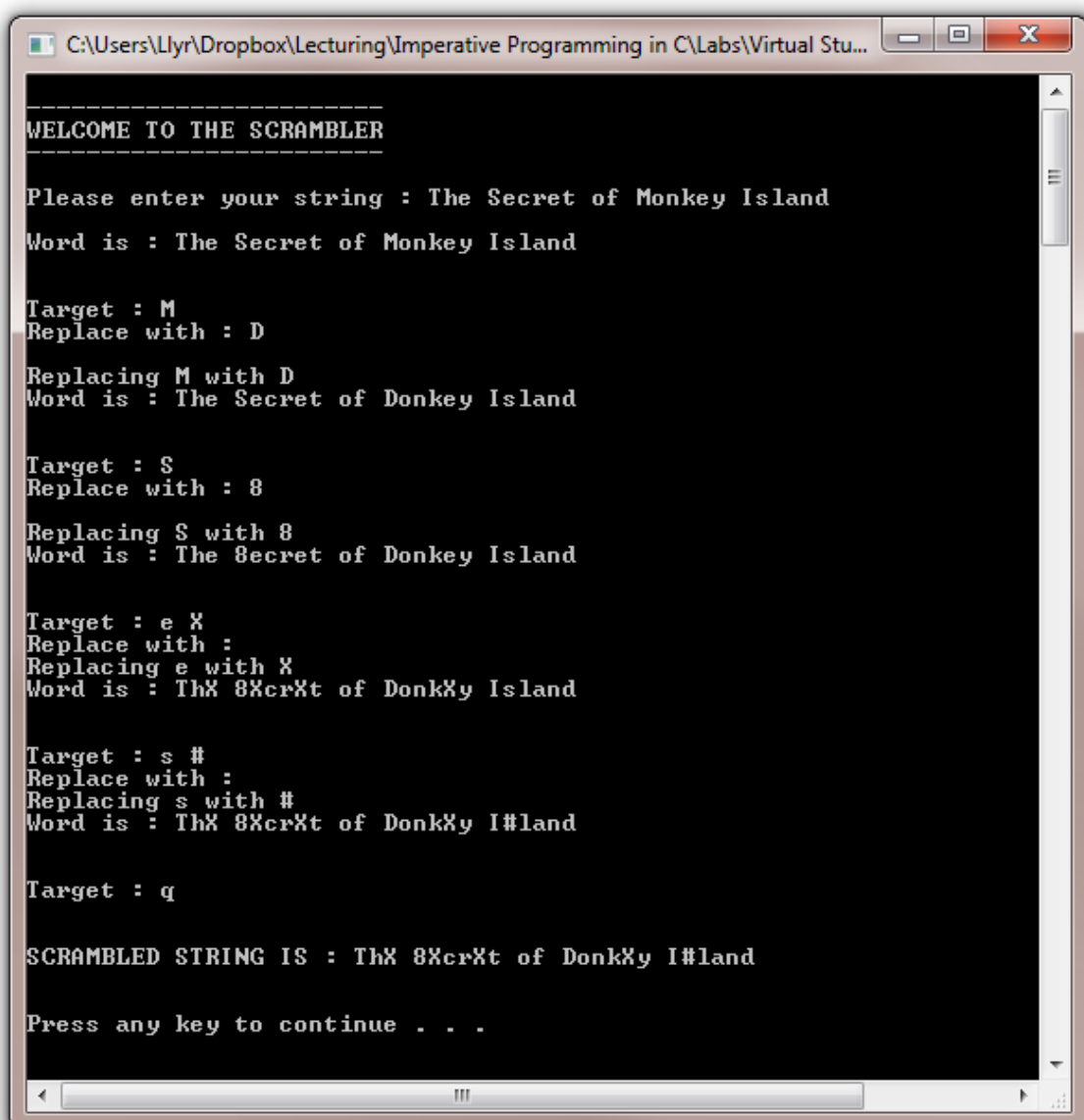
Hints

- Remember that a string is just an array of characters
- You can loop through each element in an array
- You could keep count of the vowels using an array of size 5

Assessed Work – Scrambler

Your task for this assessed work is to write a program that allows the user to scramble and unscramble a string. The program should work like this:

1. Welcome the user to the program
2. Ask the user to enter a string
3. (In a loop) Ask the user for a target character and a replacement character
4. Replace any occurrences of target char in the string with the replacement char
5. Quit when the user types in a special sentinel character as target ('q' in the example)



```
C:\Users\Llyr\Dropbox\Lecturing\Imperative Programming in C\Labs\Virtual Stu...  
-----  
WELCOME TO THE SCRAMBLER  
-----  
Please enter your string : The Secret of Monkey Island  
Word is : The Secret of Monkey Island  
  
Target : M  
Replace with : D  
Replacing M with D  
Word is : The Secret of Donkey Island  
  
Target : S  
Replace with : 8  
Replacing S with 8  
Word is : The 8ecret of Donkey Island  
  
Target : e X  
Replace with :  
Replacing e with X  
Word is : ThX 8XcrXt of DonkXy Island  
  
Target : s #  
Replace with :  
Replacing s with #  
Word is : ThX 8XcrXt of DonkXy I#land  
  
Target : q  
  
SCRAMBLED STRING IS : ThX 8XcrXt of DonkXy I#land  
  
Press any key to continue . . .
```


You scrambler should also perform the following two tasks:

1. Store a history of changes made to the string (known as the key cipher). For the previous example, the key cypher would be **MD S8 eX s#**.
2. Stop the user from using replacement characters that already exist in the string. For example:

String – “lemons”

Replace e with x

String – “lxmons”

Replace s with o.

ERROR – “o already exists in the string”

String – “lxmons”

In the above example, we cannot replace ‘s’ with ‘o’ because ‘o’ already exists in the string. This rule will ensure that information can never be destroyed, and provided there’s a key cypher we can work backwards and unscramble a string.

Once finished, your Scrambler program should be able to unscramble strings, provided you have access to the key cypher! Can you unscramble the following messages?

String : #a**wo@d1T34A

Key – s* p# r@ 2T 5A

String : l 3m 2he w3lr6s, 788 788 73 7h88

Key – o8 c7 a3 u6 t2 w4

String : l'^ .*,ing +h*.* fin* ,*a+h*r -ack*+.

Key – m^ j- t+ e* l, s.

String : Y9u'r5 39min7 t9 1 w9945n 2ri475. W85n y9u 3r9ss t85 2ri475, y9u t8r9w t85 217 6r9m t85 l56t win49w 96 t85 m9vin7 31r. Y9u'r5 25in7 w1t3854.

Key – a1 b2 c3 d4 e5 f6 g7 h8 o9

Hints

- Get the basic scrambler to work before attempting the two extended tasks
- Consider defining a test string in your code so you don't have to always type a string into the program when running
- In order to accommodate spaces in your string, consider using [fgets](#) instead of scanf.
- You can create an infinite loop with while(1). You can use the **break** keyword to exit this loop

Make sure to read the submission notes.

Submission

Use **Blackboard** to submit your source code file. For this work you only need to submit the .c file that contains your scrambler program. Ensure that the source code:

- Contains an program header (see Appendix)
- Contains an appropriate level of comments
- Follows a consistent style of indentation
- Follow the usual C programming conventions

The deadline for submission will be published on Blackboard. Late submissions will be penalised in line with School policy.

Marks for this laboratory exercise are awarded for

- Managing input and output
- Meaningful variable names
- Program correctness (i.e. do you print the correct information)
- Program testing
- Layout and structure
- Conceptual understanding

When submitting work it is your responsibility to ensure that all work submitted is

- Consistent with stated requirements
- Entirely your own work
- Submitted through Blackboard on time

Please note that there are severe penalties for submitting work which is not your own. If you have used code which you have found on the Internet or from any other source then you must signal that fact with appropriate program comments.

Appendix

[Online C Programming Resources](#)

[Complete C Reference Library](#)

C Programming IDE's

[Dev-C++](#) (Windows)

[Code::Blocks](#) (Windows, Mac, Linux)

[Visual Studio/C++ Express](#) (Windows)

[Netbeans C/C++](#) (Windows, Mac, Linux)

[Codelite](#) (Windows, Mac, Linux)



PRIFYSGOL
BANGOR
UNIVERSITY