14880673

Programing Journal

Start time: 2nd march 4 pm
finish time: 6pm
Entry 1
exe one

```
writing
 hello
        programing
                        students
```

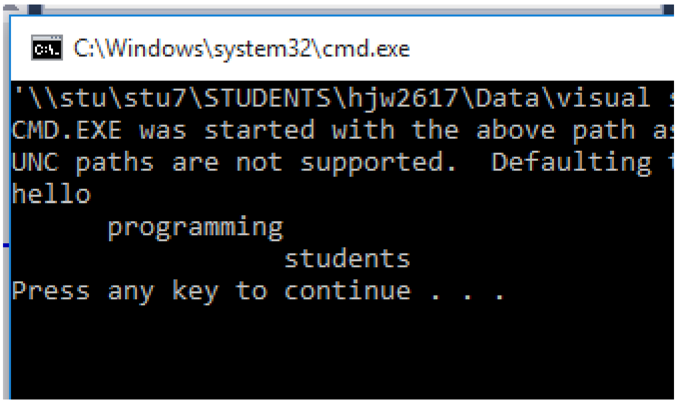
At first I tried to write the code all in one line but could not get the spacing correct so I tried separate lines i.e.

```
#include <stdio.h>
 int main()
{
	printf("hello \n");
	printf("     programming \n");
	printf("               students \n");

	return 0;
}
```

in the end i made this code to make it fit on one line

```
#include <stdio.h>
int main()
{
	printf("Hello \n    Programming \n              Students \n");
	return 0;
}
```

the program ran and shows



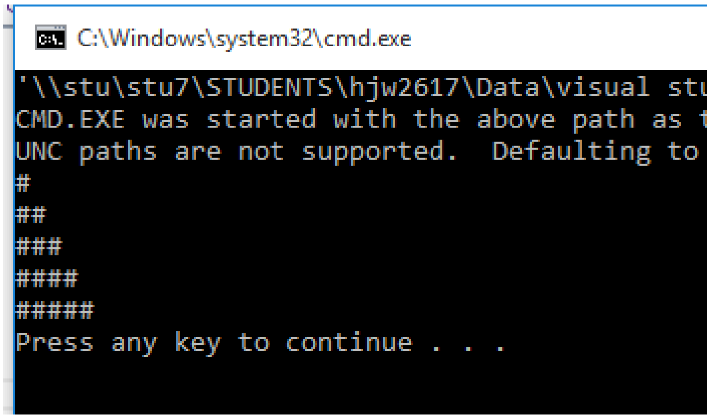
```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual s
CMD.EXE was started with the above path as
UNC paths are not supported.  Defaulting t
hello
      programming
                 students
Press any key to continue . . .
```

Exe 2

I wrote the code but I miss spelled printf I put a capital on it printF so it would not run
I also had a problem with my int main() as I added a new source not a new project had to change the other source from int main() to int main1() in order for my new source to work I wrote the code like this

```
#include <stdio.h>

int main()

{
	printf("# \n");
	printf("## \n");
	printf("### \n");
	printf("#### \n");
	printf("##### \n");

	return 0;

}
```

The output is

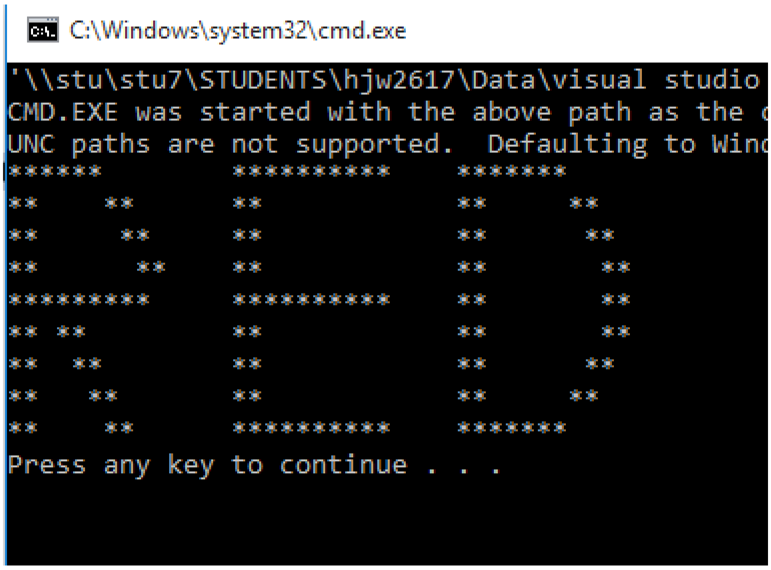


```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual stu
CMD.EXE was started with the above path as t
UNC paths are not supported.  Defaulting to
#
##
###
####
#####
Press any key to continue . . .
```

14880673
exe 3

At first I started to copy the SDH instead of putting my initials until I was told I was not reading the question after that I changed it to RED I wrote the code like this I also put studio it popped up an error and I found it and corrected it

```
#include <stdio.h>

int main()

{
     printf("******        **********    *******       \n");
     printf("**    **      **            **     **     \n");
     printf("**     **     **            **      **    \n");
     printf("**      **    **            **       **   \n");
     printf("*********     **********    **       **   \n");
     printf("** **         **            **       **   \n");
     printf("**  **        **            **      **    \n");
     printf("**   **       **            **     **     \n");
     printf("**    **      **********    *******       \n");

           return 0;
}
```

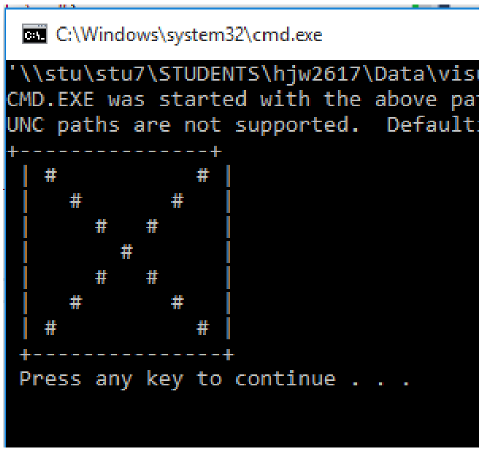


```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual studio
CMD.EXE was started with the above path as the c
UNC paths are not supported.  Defaulting to Winc
******        **********    *******
**    **      **            **     **
**     **     **            **      **
**      **    **            **       **
*********     **********    **       **
** **         **            **       **
**  **        **            **      **
**   **       **            **     **
**    **      **********    *******
Press any key to continue . . .
```

14880673
Exe4
I wrote the code with everything in alignment see below

#include <stdio.h>

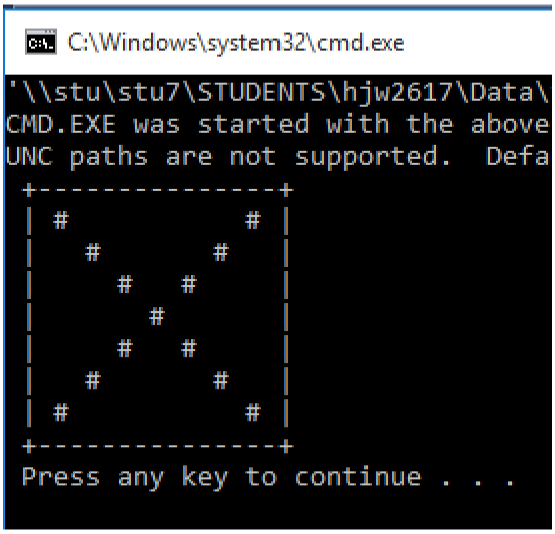
```
int main()
{
     printf("+---------------+ \n ");
     printf("| #           # | \n ");
     printf("|   #       #   | \n ");
     printf("|     #   #     | \n ");
     printf("|       #       | \n ");
     printf("|     #   #     | \n ");
     printf("|   #       #   | \n ");
     printf("| #           # | \n ");
     printf("+---------------+ \n ");
     return 0;
}
```


C:\Windows\system32\cmd.exe

```
'\\stu\stu7\STUDENTS\hjw2617\Data\vis
CMD.EXE was started with the above pa
UNC paths are not supported.  Default
+---------------+
 | #           # |
 |   #       #   |
 |     #   #     |
 |       #       |
 |     #   #     |
 |   #       #   |
 | #           # |
 +---------------+
 Press any key to continue . . .
```

but when it came to running the program its came up with the first line out off alignment
I had to change the code in include some white space at the beginning of the first line it became this

```
#include <stdio.h>

int main()
{
     printf(" +---------------+ \n ");
     printf("| #           # | \n ");
     printf("|   #       #   | \n ");
     printf("|     #   #     | \n ");
     printf("|       #       | \n ");
     printf("|     #   #     | \n ");
     printf("|   #       #   | \n ");
     printf("| #           # | \n ");
     printf("+---------------+ \n ");
     return 0;
}
```


C:\Windows\system32\cmd.exe

```
'\\stu\stu7\STUDENTS\hjw2617\Data\
CMD.EXE was started with the above
UNC paths are not supported.  Defa
 +---------------+
 | #           # |
 |   #       #   |
 |     #   #     |
 |       #       |
 |     #   #     |
 |   #       #   |
 | #           # |
 +---------------+
 Press any key to continue . . .
```

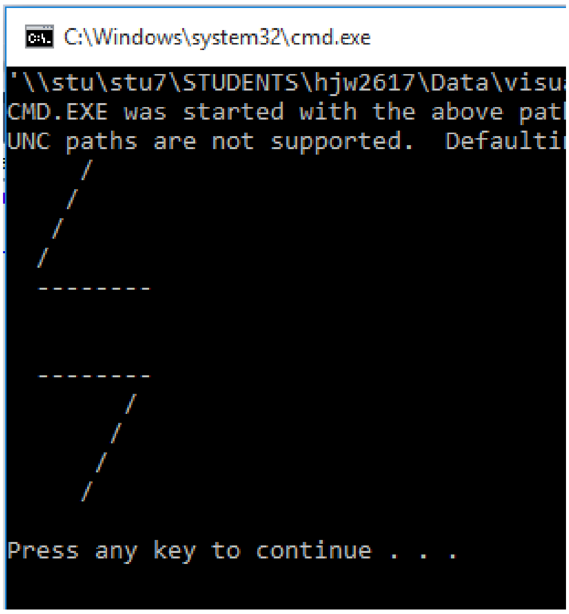
worked fine after that

all the exercises you see above I did in the lab session took approximately 2 hours

14880673
exe 5

start time:4th march 12pm
finish time: 3.30pm
entry 2
I wrote the program like as you see the code below

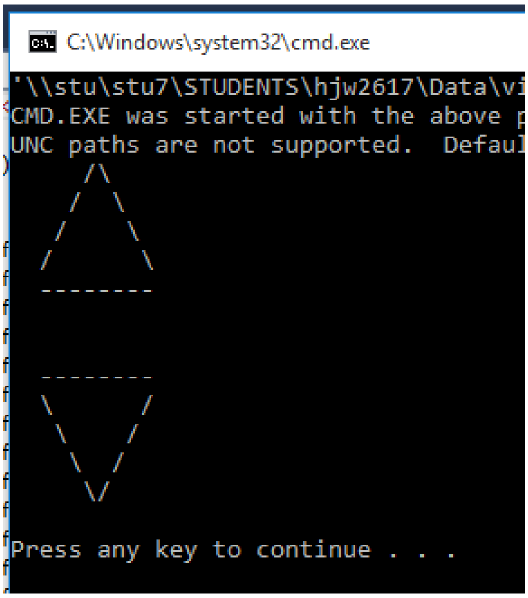
```
#include <stdio.h>

int main()

{
     printf("     /\                  \n");
     printf("    /  \                 \n");
     printf("   /    \                \n");
     printf("  /      \               \n");
     printf("  --------               \n");
     printf("                         \n");
     printf("                         \n");
     printf("  --------               \n");
     printf("  \      /               \n");
     printf("   \    /                \n");
     printf("    \  /                 \n");
     printf("     \/                  \n");
     printf("                         \n");

     return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\visu
CMD.EXE was started with the above pat
UNC paths are not supported.  Defaulti
     /
    /
   /
  /
  --------


  --------
        /
       /
      /
     /

Press any key to continue . . .
```

once I ran the program it came up with one side of each triangle missing. The side with the forward slash "\", I forgot that the "\" is a command and would not be seen as a visual unless you made it display by putting another slash in front of it, so \\ would show one \ code then became see below.

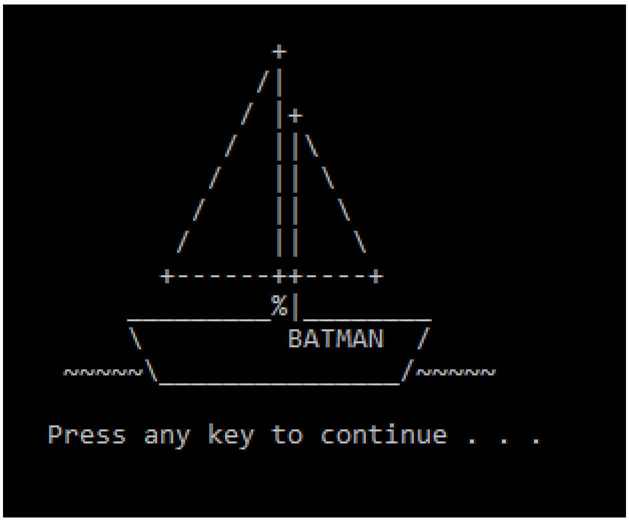
```
#include <stdio.h>

int main()

{
     printf("     /\\                 \n");
     printf("    /  \\                \n");
     printf("   /    \\               \n");
     printf("  /      \\              \n");
     printf("  --------               \n");
     printf("                         \n");
     printf("                         \n");
     printf("  --------               \n");
     printf("  \\      /              \n");
     printf("   \\    /               \n");
     printf("    \\  /                \n");
     printf("     \\/                 \n");
     printf("                         \n");

     return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\vi
CMD.EXE was started with the above p
UNC paths are not supported.  Defaul
     /\
    /  \
   /    \
  /      \
  --------


  --------
  \      /
   \    /
    \  /
     \/

Press any key to continue . . .
```

program displays a sailboat with the words BATMAN on the side. I didn’t encounter any errors when playing the program but the % would not show up on the boat, thinking that the % is once again a command in c I put a \ in front but that didn’t work. I then put one more % if front making it %% and then it showed making the code look like this.

```
#include <stdio.h>

int main()

{
      printf("                                   \n   ");
      printf("              +                    \n   ");
      printf("             /|                    \n   ");
      printf("            / |+                   \n   ");
      printf("           /  ||\\                 \n   ");
      printf("          /   || \\                \n   ");
      printf("         /    ||  \\               \n   ");
      printf("        /     ||   \\              \n   ");
      printf("       +------++----+              \n   ");
      printf("     _________%%|________          \n   ");
      printf("     \\         BATMAN  /          \n   ");
      printf(" ~~~~~\\_______________/~~~~~      \n   ");
      printf("                                   \n   ");

      return 0;
}
```



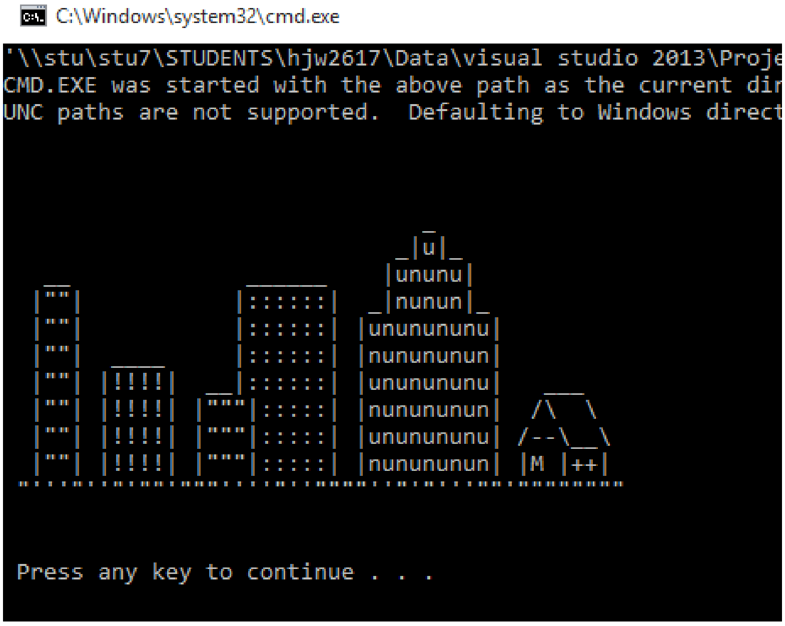
```
              +
             /|
            / |+
           /  ||\
          /   || \
         /    ||  \
        /     ||   \
       +------++----+
     _________%|________
     \         BATMAN  /
 ~~~~~\_______________/~~~~~

 Press any key to continue . . .
```

14880673
exe7

In this program it out puts the image of a city. The only real issue I came across when writing this code was making sure it lined up correctly as there were so many “\”’s moving the image across the screen. It would properly be easier to start the image from the right then go to the left as everything will move to its correct spot in the dos window without having to count spaces or lots of trial and error.

the code looks like this

```
#include <stdio.h>

int main()

{
     printf("                                                     \n ");
     printf("                                                     \n ");
     printf("                                                     \n ");
     printf("                              _                      \n ");
     printf("                            _|u|_                    \n ");
     printf("  __             ______    |ununu|                   \n ");
     printf(" |\"\"|           |::::::|  _|nunun|_                \n ");
     printf(" |\"\"|           |::::::| |ununununu|               \n ");
     printf(" |\"\"|  ____     |::::::| |nunununun|               \n ");
     printf(" |\"\"| |!!!!|  __|::::::| |ununununu|   ___         \n ");
     printf(" |\"\"| |!!!!| |\"\"\"|:::::| |nunununun|  /\\  \\   \n ");
     printf(" |\"\"| |!!!!| |\"\"\"|:::::| |ununununu| /--\\__\\  \n ");
     printf(" |\"\"| |!!!!| |\"\"\"|:::::| |nunununun| |M |++|    \n ");
     printf("\"\'\'\'\"\'\'\"\'\"\"\'\"\"\"\'\'\'\'\"\'\'\"\"\"\"\'\'\"\'\"\'\
'\'\"\"\'\"\"\"\"\"\"\"\"         \n ");
     printf("                                                     \n ");
     printf("                                                     \n ");

     return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual studio 2013\Proje
CMD.EXE was started with the above path as the current dir
UNC paths are not supported.  Defaulting to Windows direct




                               _
                             _|u|_
   __             ______    |ununu|
  |""|           |::::::|  _|nunun|_
  |""|           |::::::| |ununununu|
  |""|  ____     |::::::| |nunununun|
  |""| |!!!!|  __|::::::| |ununununu|   ___
  |""| |!!!!| |"""|:::::| |nunununun|  /\  \
  |""| |!!!!| |"""|:::::| |ununununu| /--\__\
  |""| |!!!!| |"""|:::::| |nunununun| |M |++|
 "'''"''"'""'"""''''"''""""''"'"'''""'""""""""


Press any key to continue . . .
```

Week 2
Exe 1
Start time:9th march 4pm
Finish time: 6pm
Entry 3

I wrote down some pseudo code for making this run the way I thought would work
Run program

14880673

```
Create memory locations for number1 number2 number3
Ask for input (number)
Save input to memory location number1
Ask for input (number)
Save input to memory location number2
Ask for input (number)
Save input to memory location number3
Print text “your numbers in entry order are”
Collect data from number1 number2 number3
Print number1 number2 number3 as text
Print text “your numbers in reverse order are”
Collect data from number1 number2 number3
Print data in number3 number2 number1as text
```

The first problem I ran into was the scanf error where it’s not safe and i should use the scanf_s. I fixed this by going into the property’s in the source and turning off the safety. The next problem I was running in to was how to make the numbers display, I typed printf("%d", &number1); to recall the number which did not work. I asked around my peers and found out that I didn’t need to put in the &number I can just use number1 after the %d to recall it.

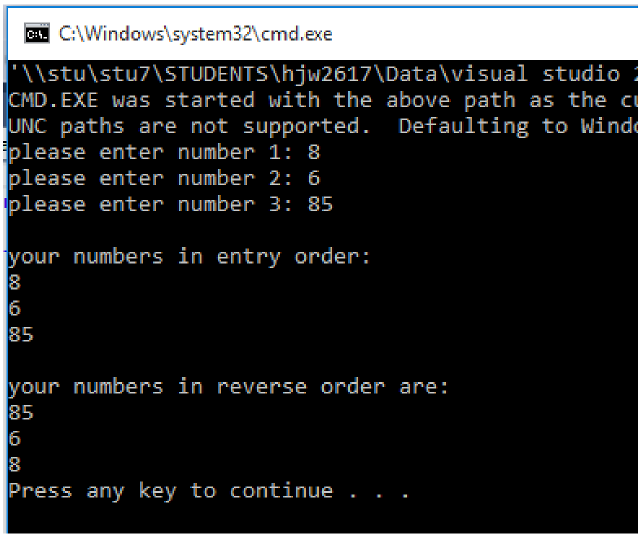
The program basically asks you to input 3 numbers in it saves then in the locations you make and then says them back to you in order entered and the reverse order.

throughout making this program I was using the debug view window to make sure that my inputted numbers were going in to the correct memory allocation.

This program took about 45 min to run though and work correctly

my program looks like this see below

```
#include <stdio.h>

int main()

{
      int number1;
      int number2;
      int number3;

      printf("please enter number 1: ");
      scanf("%d", &number1);

      printf("please enter number 2: ");
      scanf("%d", &number2);

      printf("please enter number 3: ");
      scanf("%d", &number3);
      printf("\n");

      printf("your numbers in entry order: \n");
      printf("%d \n", number1 );
      printf("%d \n", number2 );
      printf("%d \n\n", number3 );


      printf("your numbers in reverse order are: \n");
      printf("%d \n%d \n%d \n", number3, number2, number1, "\n");

      return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual studio 
CMD.EXE was started with the above path as the cu
UNC paths are not supported.  Defaulting to Windo
please enter number 1: 8
please enter number 2: 6
please enter number 3: 85

your numbers in entry order:
8
6
85

your numbers in reverse order are:
85
6
8
Press any key to continue . . .
```

14880673
Exe 2

This code gets a number from the user and saves it to the allocated memory, then uses the number entered and cubes it by multiplying it by itself 3 times and saving that into the memory named “ans”. Then it prints out the number entered and then the answer.

my first pseudo code for this was

```
run program
allocate memory number
ask for number
save input into location number
print number*number*number as text
```

when I first started this code I tried to do the maths in the printf function.

printf(“number*number*number”);

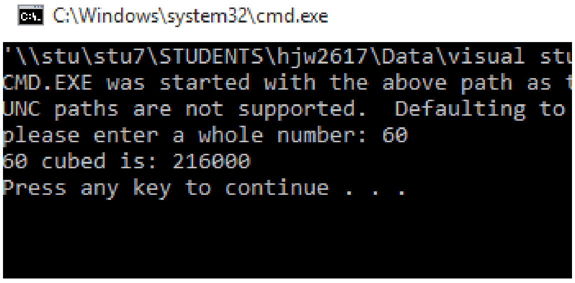
I did not make the “ans” location
I soon figured out that this was not going to work and made the “ans” location and did the maths before the printf and saved it to “ans” location.

my second pseudo code after I found out it was not going to work became.

```
run program
allocate memory number
allocate memory ans
ask for number
save number in location number
read input from location number
multiply number by number and number save to location ans
print number cubed is ans
```

the code looks like this

```
 #include <stdio.h>

int main()

{
      int number;
      int ans;

      printf("please enter a whole number: ");
      scanf("%d", &number);
      ans = number * number * number;

      printf("%d cubed is: %d \n", number,
ans);

      return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual stu
CMD.EXE was started with the above path as t
UNC paths are not supported.  Defaulting to
please enter a whole number: 60
60 cubed is: 216000
Press any key to continue . . .
```

The above 2 programs took approximately 2 hours in the lab

14880673

Exe 3

Start time: 15th march 11am
Finish time: 1.30 pm
Entry 4

Missing angle

This exercise is to get the remanding angle of a triangle using the inputs from the user.

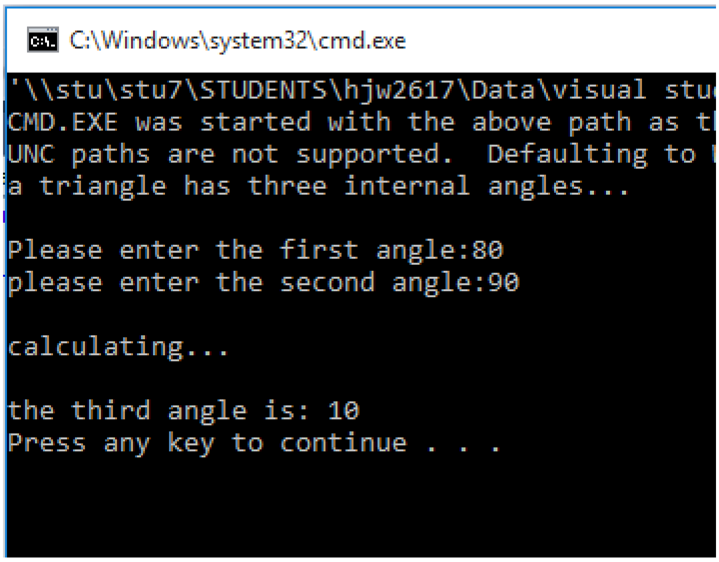
Pseudo code

```
Run

Set memory allocation num1
Set memory allocation num2
Set memory allocation num3
Set memory allocation num4
Set memory allocation ans

Ask for number
Save number in to num1

Ask for number two
Save number into num2

Set num3 to 180

Add num1 and num2 save to num4

Minus num4 from num3 save to ans

Print ans to screen
```

Using this I made this code

```c
#include <stdio.h>

int main()

{
      int num1;
      int num2;
      int num3;
      int num4;
      int ans;

      printf("a triangle has three internal angles...\n\n");
      printf("Please enter the first angle:");
      scanf(" %d", &num1);

      printf("please enter the second angle:");
      scanf("%d", &num2);

      printf("\ncalculating...\n");

      num3 = 180;
```

```
	num4 = num1 + num2;
	ans = num3 - num4;

	printf("\nthe third angle is: %d\n", ans);
		return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\visual stu
CMD.EXE was started with the above path as t
UNC paths are not supported.  Defaulting to 
a triangle has three internal angles...

Please enter the first angle:80
please enter the second angle:90

calculating...

the third angle is: 10
Press any key to continue . . .
```

looking at the code more I discovered a few redundancies that I could remove mainly the amount of memory allocations I have so the code became this.

```
#include <stdio.h>

int main()

{
	int num1;
	int num2;
	int ans;

	printf("a triangle has three internal angles...\n\n");
	printf("Please enter the first angle:");
	scanf(" %d", &num1);

	printf("please enter the second angle:");
	scanf("%d", &num2);

	printf("\ncalculating...\n");

	ans = 180 - num1 - num2;

	printf("\nthe third angle is: %d\n", ans);

	return 0;
}
```

This code took approximately 50 min to complete

14880673
Exe 4

Averaging 5 numbers

In this exercise we have to write a code that gets 5 numbers from the user and gives back the average I would say as a floating number
Pseudo code

```
Run

Setup memory allocation n1
Setup memory allocation n2
Setup memory allocation n3
Setup memory allocation n4
Setup memory allocation n5
Setup memory allocation average as a float

Print “please enter five whole numbers”

Save input into memory n1
Save input into memory n2
Save input into memory n3
Save input into memory n4
Save input into memory n5

Access memory allocation n1 print “first number: n1”
Access memory allocation n2 print “second number: n2”
Access memory allocation n3 print “third number: n3”
Access memory allocation n4 print “fourth number: n4”
Access memory allocation n5 print “fifth number: n5”

Save (n1 + n2 + n3 + n4 + n5)/5.0 into average
Print “the average of these five numbers is: average”
```

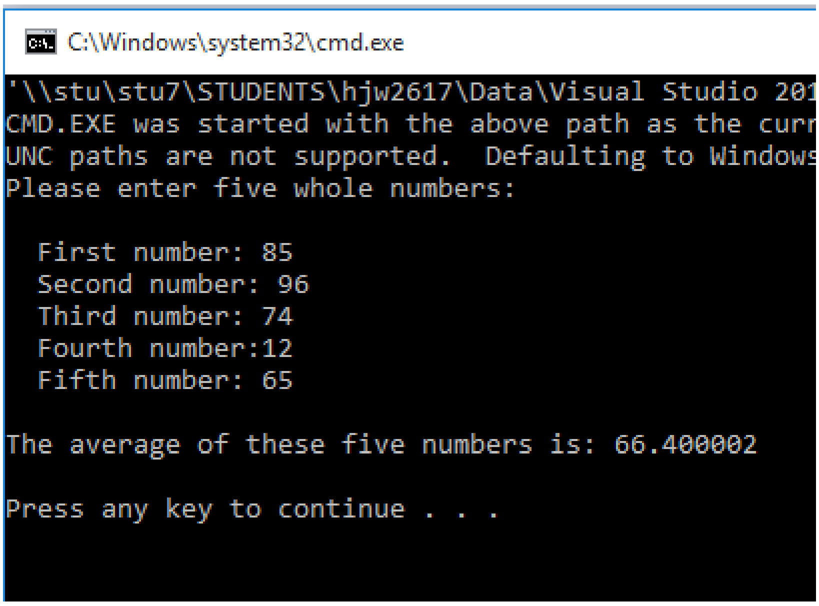
The one thing that was tripping me up in this code was when I was trying to get the average I was writing the code as

```
Average = (n1 + n2 + n3 + n4 + n5) \5
```

Which when the code ran it was popping everything back as an integer not a floating number it took me a while and some help from the skills work shop helper to find the problem I needed to divide by a floating number to get a floating answer
Also on the line above this code I forgot the ‘;’ and the end of the scanf(“%d”, &n5) so it popped back an error in the syntax thinking it was my floating memory allocation I changed it to ans
See code below

14880673

```c
#include <stdio.h>

int main()

{
	int n1;
	int n2;
	int n3;
	int n4;
	int n5;
	float ans;

	printf("Please enter five whole numbers:\n\n");

	printf("  First number: ");
	scanf("%d", &n1);

	printf("  Second number: ");
	scanf("%d", &n2);

	printf("  Third number: ");
	scanf("%d", &n3);

	printf("  Fourth number:");
	scanf("%d", &n4);

	printf("  Fifth number: ");
	scanf("%d", &n5);
	printf("\n");
		ans = (n1 + n2 + n3 + n4 + n5) / 5.0;
	printf("The average of these five numbers is: %f\n\n", ans);

	return 0;
}
```



```
'\\stu\stu7\STUDENTS\hjw2617\Data\Visual Studio 201
CMD.EXE was started with the above path as the curr
UNC paths are not supported.  Defaulting to Windows
Please enter five whole numbers:

  First number: 85
  Second number: 96
  Third number: 74
  Fourth number:12
  Fifth number: 65

The average of these five numbers is: 66.400002

Press any key to continue . . .
```

The above took approximately 1 hour to compete

Start time 15th march 2.00pm workshop
Finish time: 4 pm
Entry: 5

Week 2 exe5
In this program we are doing some more math calculating the area and the perimeter of a rectangle.

```
Pseudo code
Run

Setup memory allocation width as float
Setup memory allocation height as float
Setup memory allocation area as float
Setup memory allocation prim as float

Ask for input width save into float width
Ask for input height save into float height

Collect data from width times by 2 collect data from height times by 2 add both and save into prim
Collect data from width and times by data from height save into area

Print “the rectangles perimeter is: prim “
Print “the rectangles area is: area”
```

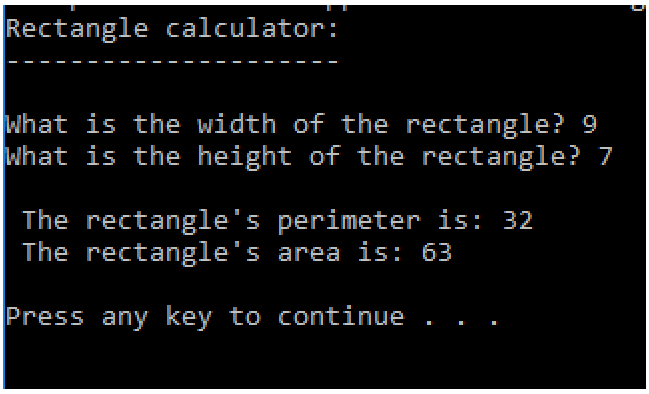
Some problems I faced in this one was when I was saving my input to the float and when I as recalling it as a float it was giving all the decimals as well. So I changed it to saving as a int not a float but when recalling that as a float “ %f” as it was in my code it would return 0, after that I changed it to a “%d” to jut recall it as an int.

I also got a crash I forgot to put the &width when using the scanf function when I got this crash I knew what it was as I’ve forgotten to put the & before.

Last problem was a simple one I was getting the bedmas wrong when it came to calculating the perimeter.

Code below

```c
#include <stdio.h>

int main()

{
	float width;
	float height;
	int prim;
	int area;

	printf("Rectangle calculator:\n");
	printf("---------------------\n\n");
printf("What is the width of the rectangle? ");
scanf("%f", &width);

printf("What is the height of the rectangle? ");
scanf("%f", &height);

prim = (width + height) * 2;
area = width * height;

printf("\n The rectangle's perimeter is: %d\n", prim);
printf(" The rectangle's area is: %d\n\n", area);

return 0;
}
```



```
Rectangle calculator:
---------------------

What is the width of the rectangle? 9
What is the height of the rectangle? 7

 The rectangle's perimeter is: 32
 The rectangle's area is: 63

Press any key to continue . . .
```

This program took approximately 30 min

14880673
Week 2 exe 6
Math operation test

In this program is going to ask for 2 whole numbers and then do some math calculations add, subtract, multiply and divide with integer division and floating point division.

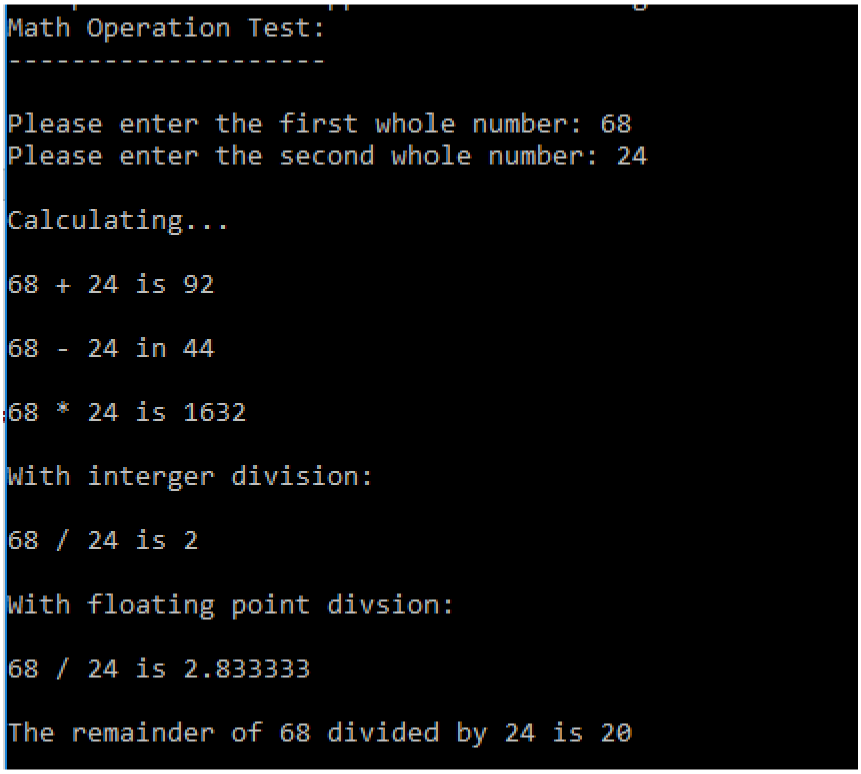
```
Pseudo code
Run
Setup memory allocation int n1
Setup memory allocation int n2
Setup memory allocation int ans
Setup memory allocation float fans
Setup memory allocation float a
Setup memory allocation float b

Ask for input save in n1
Ask for input save in n2
Save int n1 to float a
Save int n2 to float b

Get data from n1 and n2 add together save in ans

Print " n1 + n2 is ans"

Get data from n1 and n2 subtract them save into ans

Print " n1 + n2 is ans"

Get data from n1 and n2 multiply together save into ans

Print "n1 + n2 is ans"

Print "with integer division:"

Get data from n1 and n2 divide and save into ans

Print " n1 / n2 is ans"

Print "with floating point division:"

Get data from a and b divide save into fans

Print "n1 / n2 is fans"

Get data from n1 and n2 ans = n1 % n2
Print "The remainder of n1 divided by n2 is ans"
```

The main problem I had with this one is the modulus I could not find out how to get the remainder and show it. I tried multiplying numbers and dividing numbers to get the reminder went over lecture notes and found the % command but could not apply it correctly I google how to and found. [1]
The code on this page was all I needed to click as to how to apply it.

14880673
Used the code
Ans = n1 % n2;
To give me the remainder

Code below

```
#include <stdio.h>

int main()

{

    int n1;
    int n2;
    int ans;
    float a;
    float b;
    float fans;

    printf("Math Operation Test:\n");
    printf("--------------------\n\n");

    printf("Please enter the first whole number: ");
    scanf("%d", &n1);
    a = n1;

    printf("Please enter the second whole number: ");
    scanf("%d", &n2);
    b = n2;

    printf("\nCalculating...\n\n");

    ans = n1 + n2;
    printf("%d + %d is %d\n\n", n1, n2, ans);

    ans = n1 - n2;
    printf("%d - %d in %d\n\n", n1, n2, ans);

    ans = n1* n2;
    printf("%d * %d is %d\n\n", n1, n2, ans);

    printf("With interger division:\n\n");
    ans = n1 / n2;

    printf("%d / %d is %d\n\n", n1, n2, ans);
    printf("With floating point division:\n\n");
    fans = a / b;

    printf("%d / %d is %f\n\n", n1, n2, fans);

    ans = n1 %n2;

    printf("The remainder of %d divided by %d is %d\n\n", n1, n2, ans);

    return 0;

}
```

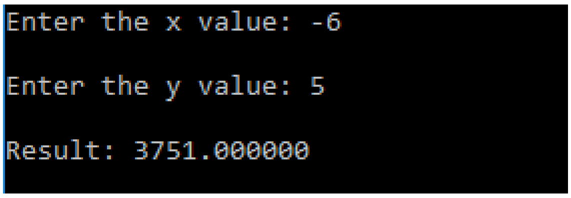


```
Math Operation Test:
--------------------

Please enter the first whole number: 68
Please enter the second whole number: 24

Calculating...

68 + 24 is 92

68 - 24 in 44

68 * 24 is 1632

With interger division:

68 / 24 is 2

With floating point divsion:

68 / 24 is 2.833333

The remainder of 68 divided by 24 is 20
```

This program took me approximately 1 hour 45 min mainly due to the modulus

14880673
Exe7
Start time: 15th march 7pm
Finish time: 9pm
Entry: 6
Pseudo code to source code

```
#include <stdio.h>

int main()

{
	int x;
	int y;
	float a;
	float b;
	float result;

	printf("Enter the x value: ");
	scanf("%d", &x);
	printf("\n");
	printf("Enter the y value: ");
	scanf("%d", &y);
	printf("\n");
	a = x * y;
	b = x + y;
	result = b * b + a * (b - x) * (a + y);

	printf("Result: %f\n\n", result);

	return 0;
}
```



```
Enter the x value: -6

Enter the y value: 5

Result: 3751.000000
```

In this code the only problem I encountered was when I used the code “scanf(“%d\n”, &x);” when running the code every time you press enter it will add a new line stopping the program from continuing.
this code took me 10 min approximately

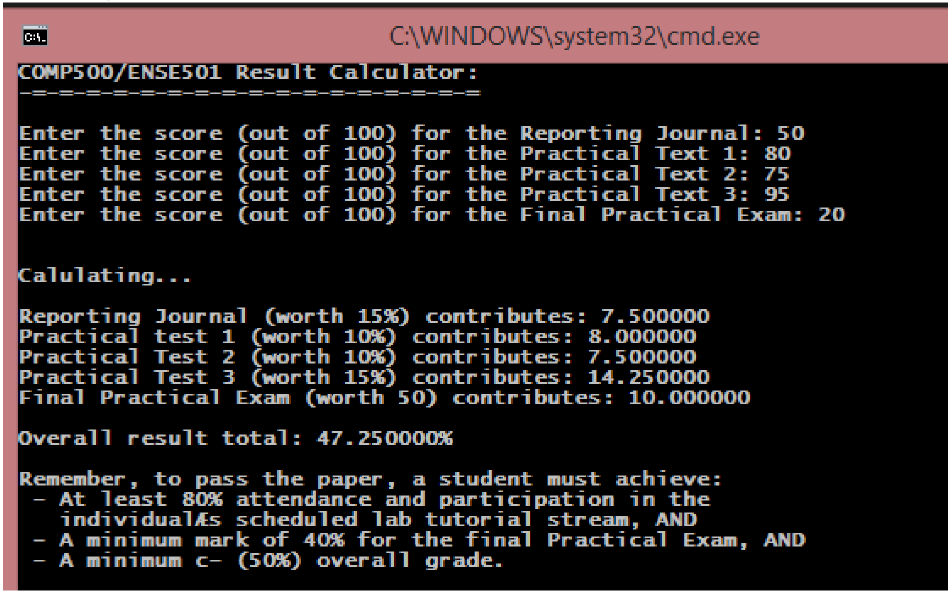
14880673
exe 8
result calculator
This program takes the inputs from the user and works out how much it contributes to the final result. The only problem I faced with this code was working out how to get the percentage of the input, due to week one when it came to showing the % sign I remembered you needed 2 % “%%” to get one to show

```
pseudo code
create memory allocations rj, pt1, pt2, pt3, exam, final as float
ask for a score out of 100 save into rj
ask for a score out of 100 save into pt1
ask for a score out of 100 save into pt2
ask for a score out of 100 save in to pt3
ask for a score out of 100 save into exam

collect data from rj and multiply by .15 save to rj
collect data from pt1 and multiply by .1 save to pt1
collect data from pt2 and multiply by .1 save to pt2
collect data from pt3 and multiply by .15 save to pt3
collect data from exam and multiply by .5 save to exam

print Calculating

print Reporting journal (worth 15%) contributes: rj
print Practical test 1 (worth 10%) contributes: pt1
print Practical test 2 (worth 10%) contributes: pt2
print Practical test 3(worth 15%) contributes: pt3
print Final Practical Exam (worth 50%) contribute: exam

collect data from rj, pt1, pt2, pt3, exam and add together save to final

print Overall result total: final

print Remember, to pass the paper, a student must achieve:
print  - at least 80% attendance and participation in the individual’s scheduled lab tutorial stream,
And
print  - A minimum mark of 40% for the final practical Exam, AND
print -  A minimum c- (50%) overall grade.
```

Code below

```c
#include <stdio.h>

int main()

{
	float rj, pt1, pt2, pt3, exam, final;
	printf("COMP500/ENSE501 Result Calculator:\n");
	printf("-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n");
	printf("Enter the score (out of 100) for the Reporting Journal: ");
	scanf("%f", &rj);
	printf("Enter the score (out of 100) for the Practical Text 1: ");
	scanf("%f", &pt1);
	printf("Enter the score (out of 100) for the Practical Text 2: ");
	scanf("%f", &pt2);
	printf("Enter the score (out of 100) for the Practical Text 3: ");
	scanf("%f", &pt3);
	printf("Enter the score (out of 100) for the Final Practical Exam: ");
	scanf("%f", &exam);

	rj = rj * 0.15;
	pt1 = pt1 *0.1;
	pt2 = pt2 *0.1;
	pt3 = pt3 *0.15;
	exam = exam *.50;

	final = rj + pt1 + pt2 + pt3 + exam;

	printf("\n\nCalulating...\n\n");

	printf("Reporting Journal (worth 15%%) contributes: %f\n", rj);
	printf("Practical test 1 (worth 10%%) contributes: %f\n", pt1);
	printf("Practical Test 2 (worth 10%%) contributes: %f\n", pt2);
	printf("Practical Test 3 (worth 15%%) contributes: %f\n", pt3);
	printf("Final Practical Exam (worth 50%) contributes: %f\n\n", exam);

	printf("Overall result total: %f%%\n\n", final);
	printf("Remember, to pass the paper, a student must achieve:\n");
	printf(" - At least 80%% attendance and participation in the\n
individual's scheduled lab tutorial stream, AND\n");
	printf(" - A minimum mark of 40%% for the final Practical Exam, AND\n");
	printf(" - A minimum c\- (50%%) overall grade.\n\n");



	return 0;
```

this program took approximately 1 hour.



```
COMP500/ENSE501 Result Calculator:
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Enter the score (out of 100) for the Reporting Journal: 50
Enter the score (out of 100) for the Practical Text 1: 80
Enter the score (out of 100) for the Practical Text 2: 75
Enter the score (out of 100) for the Practical Text 3: 95
Enter the score (out of 100) for the Final Practical Exam: 20


Calulating...

Reporting Journal (worth 15%) contributes: 7.500000
Practical test 1 (worth 10%) contributes: 8.000000
Practical Test 2 (worth 10%) contributes: 7.500000
Practical Test 3 (worth 15%) contributes: 14.250000
Final Practical Exam (worth 50) contributes: 10.000000

Overall result total: 47.250000%

Remember, to pass the paper, a student must achieve:
 - At least 80% attendance and participation in the
   individualÆs scheduled lab tutorial stream, AND
 - A minimum mark of 40% for the final Practical Exam, AND
 - A minimum c- (50%) overall grade.
```

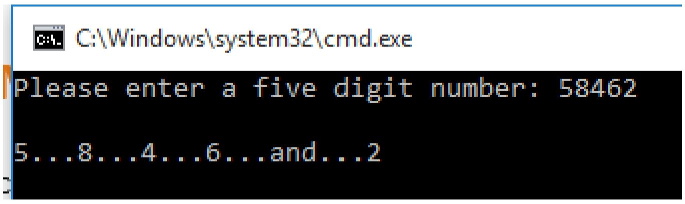
exe9
number separation:
In this program it takes the inputs from the user as numbers then separates them and recalls them one by one. Rather than saving them exactly as inputted the program saves each character in a separate location.

Before I started writing this I was trying to work out the math on how to get the one number and save them to different locations. I thought there must be a better way so I looked on the internet for the faster more effective way to save the numbers, I found that I could save them as “char”. And save one number to each car location I used this website to help with the code. [2]

```
Pseudo code
Run
Create char location, 1, 2, 3, 4 ,5

Print “Please enter a five digit number: “
save each character in locations 1 ,2, 3, 4, 5

collect data from locations 1, 2, 3, 4,5
Print “1…2…3…4…and…5”
```

```c
int main()

{

	char n1, n2, n3, n4, n5;

	printf("Please enter a five digit number: ");
	scanf("%c%c%c%c%c", &n1, &n2, &n3, &n4, &n5);
	printf("\n%c...%c...%c...%c...and...%c\n\n", n1, n2, n3, n4, n5);

	return 0;
}
```

This program took approximately 50 min.

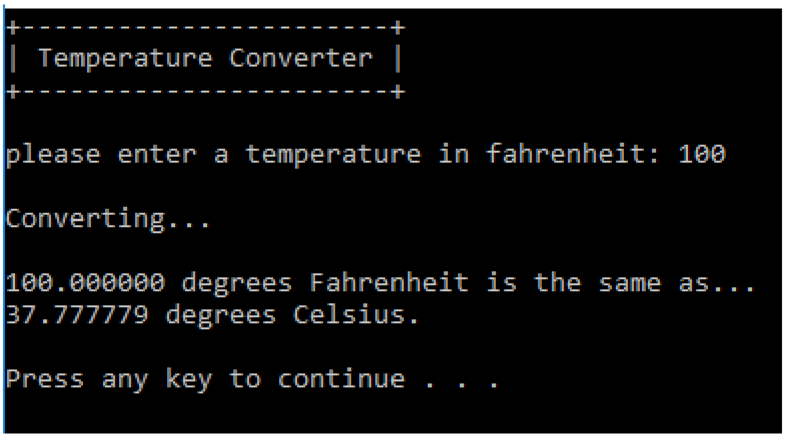


```
Please enter a five digit number: 58462

5...8...4...6...and...2
```

14880673
Start time:16th march 4pm
Finish time: 6pm
Entry: 7
week 3 exe1

In this program we are using the input and converting it to a Celsius using math.
code as below nothing really went wrong except when I first ran the program I forgot to call the variable ans

```
#include <stdio.h>

int main()

{
	float temp, ans;

	printf("+-----------------------+\n");
	printf("| Temperature Converter |\n");
	printf("+-----------------------+\n\n");

	printf("please enter a temperature in fahrenheit: ");
	scanf("%f", &temp);
	printf("\nConverting...\n");

	ans = (5.0 / 9.0) * (temp - 32.0);

	printf("\n%f degrees Fahrenheit is the same as...\n", temp);
	printf("%f degrees Celsius.\n\n", ans);

	return 0;
}
```



```
+-----------------------+
| Temperature Converter |
+-----------------------+

please enter a temperature in fahrenheit: 100

Converting...

100.000000 degrees Fahrenheit is the same as...
37.777779 degrees Celsius.

Press any key to continue . . .
```

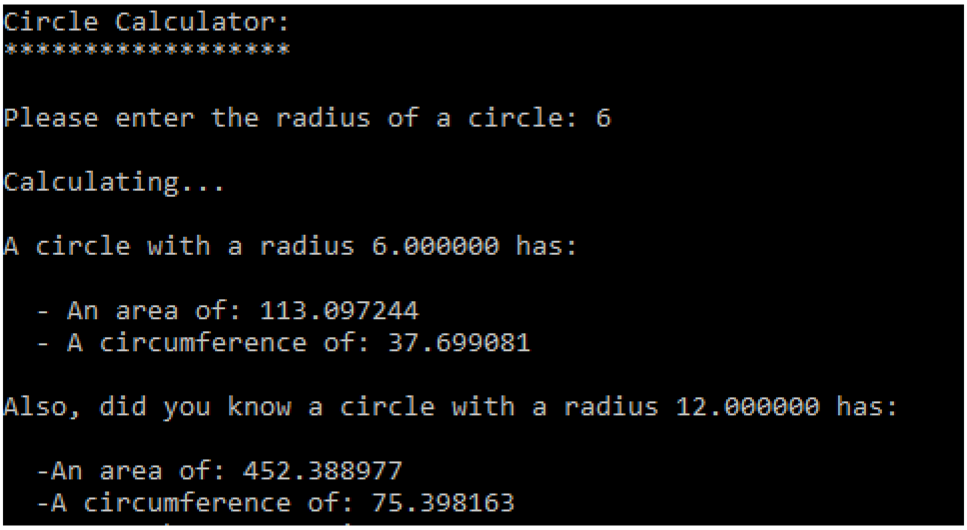
This code took me approximately 30 min to complete

.

Week 3 exe 2
Circle calculator
This program uses the inputs given by the user to calculate the radius of the circle and display it.
For this one I felt I did not need to write a pseudo code, I did discover that not having writing it took longer to code as I had to make sure the steps where in the correct order.

```
#include <stdio.h>

int main()
{
    float n1, area1, circ1, ans, area2, circ2;

    printf("Circle Calculator:\n");
    printf("******************\n\n");

    printf("Please enter the radius of a circle: ");
    scanf("%f", &n1);

    ans = n1 * 2;
    area1 = 3.14159f * (n1*n1);
    circ1 = 2 * 3.14159f*n1;
    area2 = 3.14159f *(ans*ans);
    circ2 = 2 * 3.14159f * ans;

    printf("\nCalculating...\n\n");
    printf("A circle with a radius %f has:\n\n", n1);
    printf("  - An area of: %f\n", area1);
    printf("  - A circumference of: %f\n\n", circ1);
    printf("Also, did you know a circle with a radius %f has:\n\n", ans);
    printf("  -An area of: %f\n", area2);
    printf("  -A circumference of: %f\n", circ2);

    return 0;

}
```

The program took approximately 1 hour and half to write.



```
Circle Calculator:
******************

Please enter the radius of a circle: 6

Calculating...

A circle with a radius 6.000000 has:

  - An area of: 113.097244
  - A circumference of: 37.699081

Also, did you know a circle with a radius 12.000000 has:

  -An area of: 452.388977
  -A circumference of: 75.398163
```

14880673
Start time:17th march 11am
Finish time: 1pm
Entry: 8
Week 3 exe 3

Alphabet facts:
In this code it takes the input from the user and saves it into a location as an ascii value. Then using the ascii table it adds and subtracts numbers to get to the corresponding value.

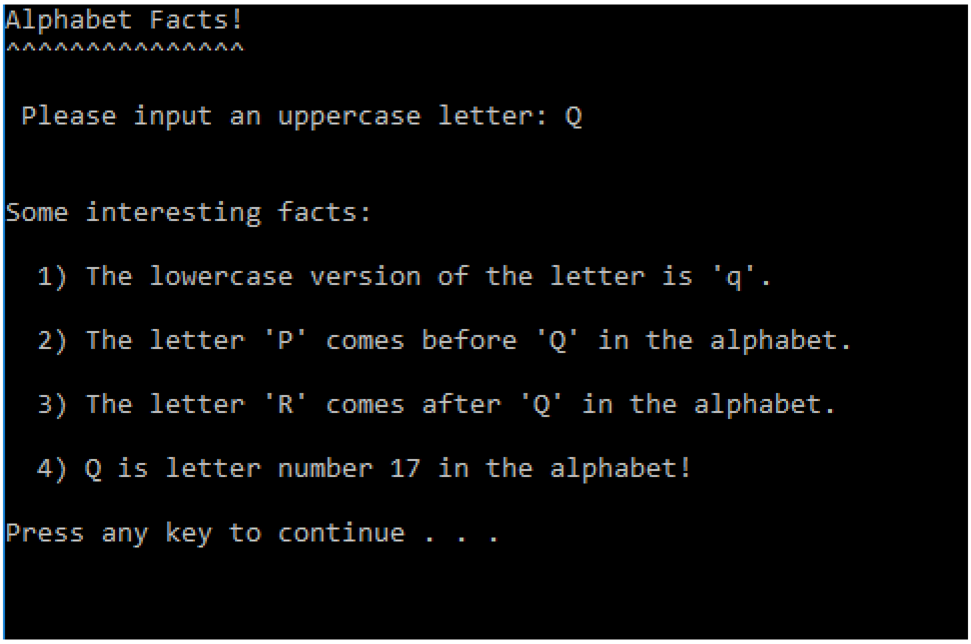
```
Pusdo code
Run

Set allocations n1, n2, n3, n4, n5 as char
Ask for input save to location n1

n2 = n1 +32
n3 = n1 – 1
n4 = n1 + 1
n5 = n1 – 64

print “some interesting facts:”
print  “1) the lowercase version of the letter is n2”
print “ 2) the letter n3 comes before n2 in the alphabet.”
Print “ 3) the letter n4 comes after n2 in the alphabet.”
Print “4) n1(as char) is letter number n5 (as int) in the alphabet! “
```

Some problems I found was how to get the number in the alphabet to show. At first I tried taking away 18 from the number, but that would only give you correct answers up to letter 9. I then remembered that you can recall ascii numbers as int not just char. Then I subtracted 65 from the number which seemed to work fine until I tested “A” and it came back with A is number 0 in the alphabet so I changed it to -64 and it worked fine.
Another problem I have come across is when you input ‘A’ or ‘Z’ the letter before or after will go to the next ascii code value ie ‘@ comes before A in the alphabet’ and ‘[comes after Z in the alphabet’.
I looked on the internet for a way to fix this problem and I found if else statements and put that in the code
Code below is the lab exercise

```c
#include <stdio.h>
int main()
{
	char n1, n2, n3, n4, n5;
	printf("Alphabet Facts!\n");
	printf("^^^^^^^^^^^^^^^\n\n");

	printf(" Please input an uppercase letter: ");
	scanf("%c", &n1);

	n2 = n1 + 32;
	n3 = n1 - 1;
	n4 = n1 + 1;
	n5 = n1 - 64;

	printf("\n\nSome interesting facts:\n\n");
	printf("  1) The lowercase version of the letter is '%c'.\n\n", n2);
	printf("  2) The letter '%c' comes before '%c' in the alphabet.\n\n", n3,
n1);
	printf("  3) The letter '%c' comes after '%c' in the alphabet.\n\n", n4,
n1);
	printf("  4) %c is letter number %d in the alphabet!\n\n", n1, n5);
```

```
    return 0;
}
```

this was the out put

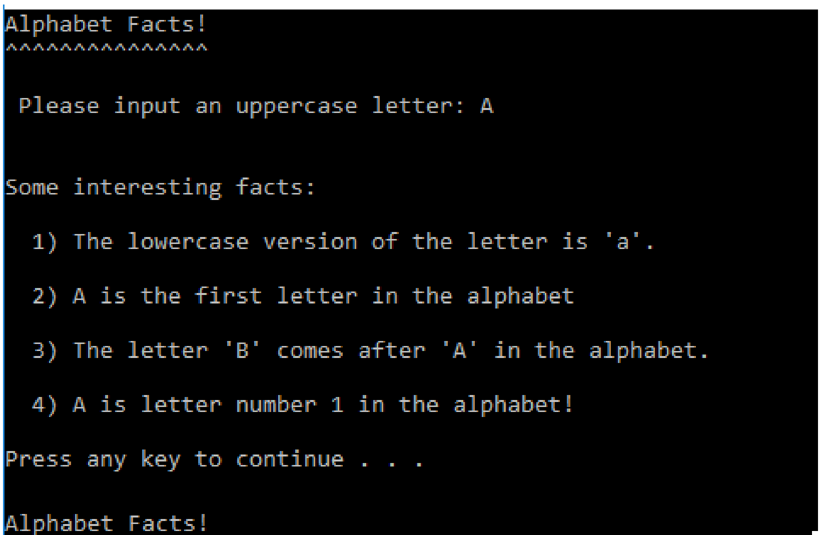


```
Alphabet Facts!
^^^^^^^^^^^^^^^

 Please input an uppercase letter: Q


Some interesting facts:

  1) The lowercase version of the letter is 'q'.

  2) The letter 'P' comes before 'Q' in the alphabet.

  3) The letter 'R' comes after 'Q' in the alphabet.

  4) Q is letter number 17 in the alphabet!

Press any key to continue . . .
```

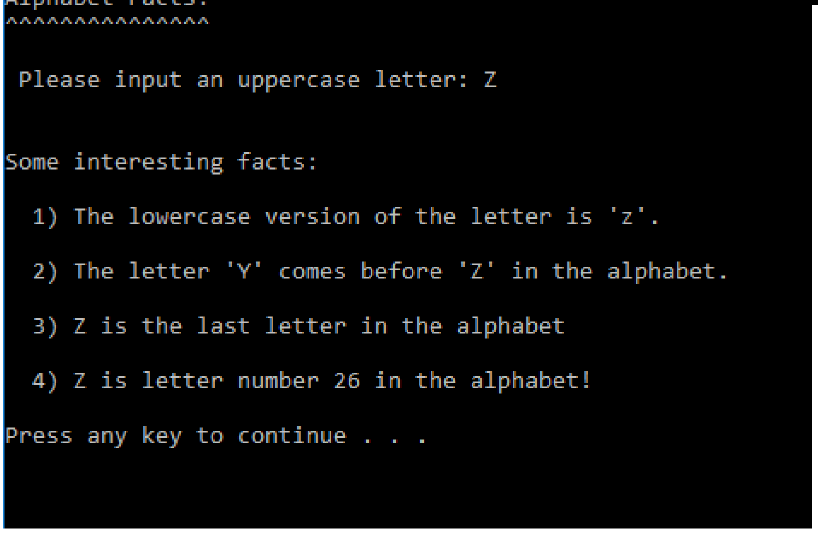
This program took me approximately 2 hours most of that time was trying to discover how to get the number in the alphabet to display correctly
Code below is modified to allow for if else statements

```
#include <stdio.h>
int main()
{
    char n1, n2, n3, n4, n5;
    printf("Alphabet Facts!\n");
    printf("^^^^^^^^^^^^^^^\n\n");

    printf(" Please input an uppercase letter: ");
    scanf("%c", &n1);

    n2 = n1 + 32;
    n3 = n1 - 1;
    n4 = n1 + 1;
    n5 = n1 - 64;

    printf("\n\nSome interesting facts:\n\n");
    printf("  1) The lowercase version of the letter is '%c'.\n\n", n2);
    if (n1 == 65)
    {
        printf("  2) %c is the first letter in the alphabet\n\n", n1);
    }
    else
    {
        printf("  2) The letter '%c' comes before '%c' in the
alphabet.\n\n", n3, n1);
    }
    if (n1 == 90)
    {
        printf("  3) %c is the last letter in the alphabet\n\n", n1);
    }
```

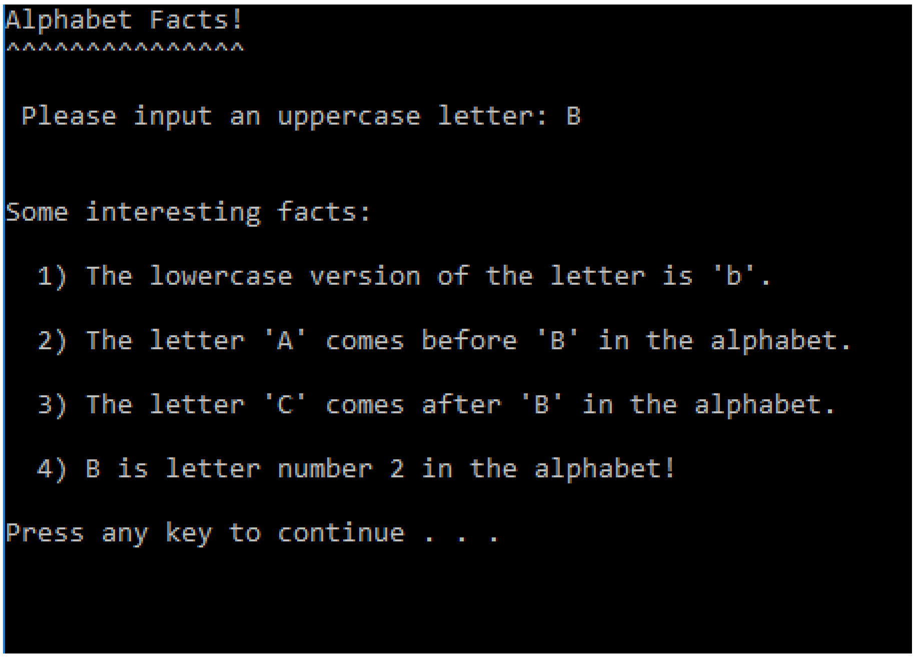
```
       else
       {
              printf("  3) The letter '%c' comes after '%c' in the alphabet.\n\n",
n4, n1);
       }
       printf("  4) %c is letter number %d in the alphabet!\n\n", n1, n5);

       return 0;
}
```



```
Alphabet Facts!
^^^^^^^^^^^^^^^

 Please input an uppercase letter: A


Some interesting facts:

  1) The lowercase version of the letter is 'a'.

  2) A is the first letter in the alphabet

  3) The letter 'B' comes after 'A' in the alphabet.

  4) A is letter number 1 in the alphabet!

Press any key to continue . . .
```



```
Alphabet Facts!
^^^^^^^^^^^^^^^

 Please input an uppercase letter: Z


Some interesting facts:

  1) The lowercase version of the letter is 'z'.

  2) The letter 'Y' comes before 'Z' in the alphabet.

  3) Z is the last letter in the alphabet

  4) Z is letter number 26 in the alphabet!

Press any key to continue . . .
```



```
Alphabet Facts!
^^^^^^^^^^^^^^^

 Please input an uppercase letter: B


Some interesting facts:

  1) The lowercase version of the letter is 'b'.

  2) The letter 'A' comes before 'B' in the alphabet.

  3) The letter 'C' comes after 'B' in the alphabet.

  4) B is letter number 2 in the alphabet!

Press any key to continue . . .
```

Modifying the code took about one hour to find and apply I used [3] website to help me with this.

14880673
Start time: 22/march 4:25pm
Finish time: 8.50 pm
Entry: 9
Week 4 exe 4

Code

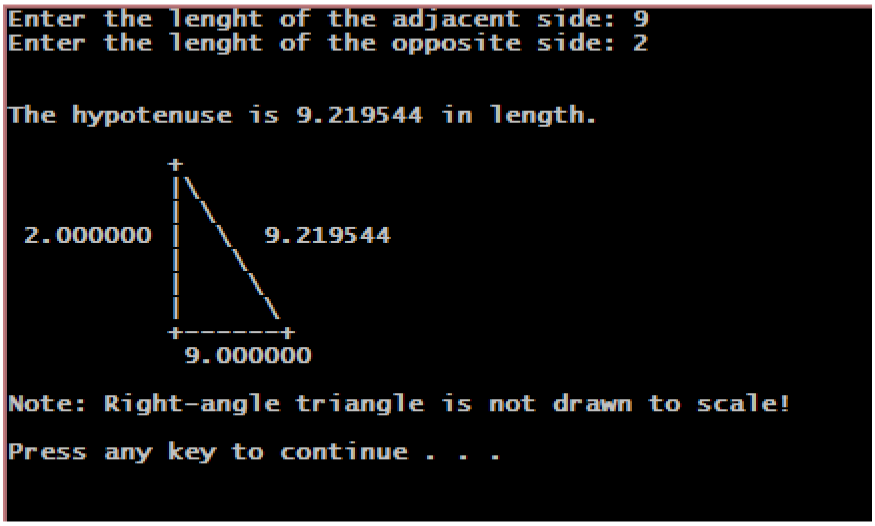
```
Set up memory allocations n1, n2, ans as floats

Ask user for input save into n1
Ask user for input save into n2

Collect data from n1 and n2 square root (n1^2 + n2^2) save into ans

Print the hypotenuse is ans in length

Create display of a triangle and ad n1 n2 and ans in their appropriate locations

Print note the right-angle is not drawn to scale
End
```

This program collects the user input does some Pythagorean theorem and out puts the hypotenuse

The only problem I found whilst writing the code was how to do the square root looked over the slides and used the example code math square root: turns out I had forgotten to add the # include <math.h> to the code

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

int main()
{
     float n1, n2, ans;
     printf("Enter the lenght of the adjacent side: ");
     scanf("%f",&n1);
     printf("Enter the lenght of the opposite side: ");
     scanf("%f", &n2);

     ans = sqrt(n1*n1 + n2*n2);
     printf("\n\nThe hypotenuse is %f in length.\n\n", ans);

     printf("          +\n");
     printf("          |\\\n");
     printf("          | \\\n");
     printf(" %f |  \\  %f \n",n2, ans);
     printf("          |   \\\n");
     printf("          |    \\\n");
     printf("          |     \\\n");
     printf("          +------+\n");

     printf("           %f \n\n", n1);

     printf("Note: Right-angle triangle is not drawn to scale!\n\n");

}
```

Out put below



```
Enter the lenght of the adjacent side: 9
Enter the lenght of the opposite side: 2


The hypotenuse is 9.219544 in length.

          +
          |\
          | \
 2.000000 |  \  9.219544
          |   \
          |    \
          |     \
          +-------+
           9.000000

Note: Right-angle triangle is not drawn to scale!

Press any key to continue . . .
```

Exe 5
D20 Dice Roller:

Code
Run

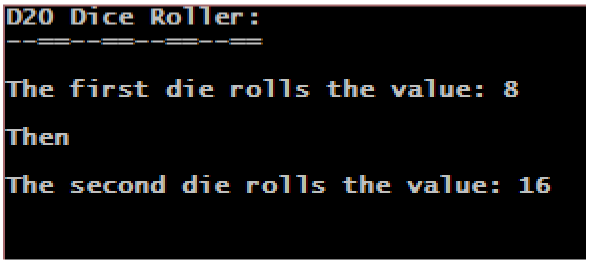
```
#include <stdio.h>
#include <stdlib.h>
#include< time.h>

Set up the seed srand( time ());

Set up int roll1, roll2,

Print D20 dice roller:
Print --==--==--==--==

Roll1 = (rand() % 20 + 1;
Roll2 = (rand() % 20 + 1;

Print the first die rolls the value: roll1
Print then
Print the second die rolls the value: roll2

End
```

This is a random number generator it sets up the seed under the time and rolls twice and sets saves it in the ints called roll1 and roll2 then prints the outcome

The main issue I had with this one was I didn't put in <stdio.h> thinking that I didn't need to because I was not using scanf input I forgot it was for output as well

Code below

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
int main()
{
	srand(time(0));
	printf("D20 Dice Roller:\n");
	printf("--==--==--==--==\n\n");

	int roll1 = (rand() % 20) + 1;
	int roll2 = (rand() % 20) + 1;


	printf("The first die rolls the value: %d\n\n", roll1);
	printf("Then\n\n");
	printf("The second die rolls the value: %d\n\n\n\n", roll2);

}
```



```
D20 Dice Roller:
--==--==--==--==

The first die rolls the value: 8

Then

The second die rolls the value: 16
```

Week 4 exe 6

Average of five using array:
Code
Run
Setup array [5]

Ask for 5 real numbers
Save in the arrays 1 to 5

Print “ the user entered:”

Gather data from the arrays and print

Add all the numbers together and divide by 5 save in to ans
Print “the average of these five numbers is ans

This program asks the user for 5 numbers and saves them into the arrays says the inputs back to the user and gets the average of the 5 numbers and displays that at the end

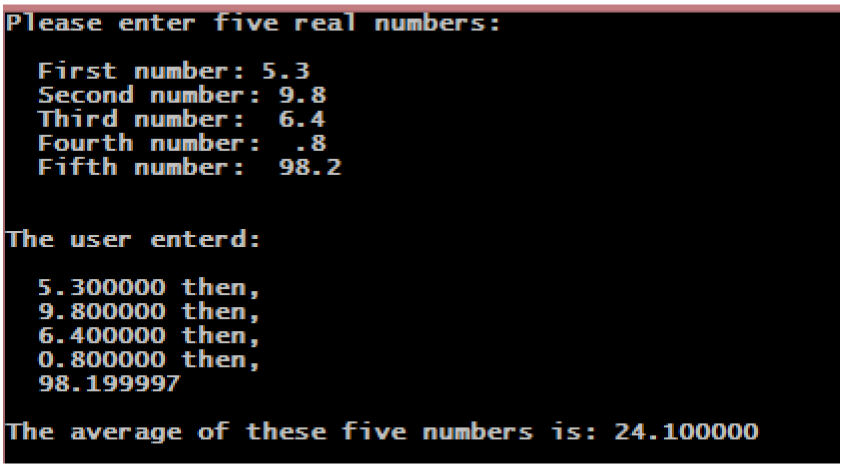
Writing the program for this I got to the end and ran the program and everything went well the code complied and ran until the end then it crashed over and over giving me the error ”Run-Time Check Failure #2 - Stack around the variable 'num' was corrupted.” It took me over an hour to find out why this was happening going through and debugging and putting in break points trying to find the error in the code. The error kept happening after at the “return 0;” command.
I was looking at the example codes and still could not find it out. Went away for dinner came back had another crack at it and clicked that programmers count from zero not one
Looking at my code again I set up the array as num[5] but I was saving and calling from arrays num[1]-[5]

I changed them to save from num[0]-[4] and it ran smoothly.

Code below

```c
#include <stdio.h>

int main()

{
     float num[5];
     float ans;

     printf("Please enter five real numbers:\n\n");

     printf("  First number: ");
     scanf("%f", &num[0]);
     printf("  Second number: ");
     scanf("%f", &num[1]);
     printf("  Third number:  ");
     scanf("%f", &num[2]);
     printf("  Fourth number:  ");
     scanf("%f", &num[3]);
     printf("  Fifth number:  ");
     scanf("%f", &num[4]);

     printf("\n\nThe user enterd:\n\n");

     printf("  %f then,\n", num[0]);
     printf("  %f then,\n", num[1]);
     printf("  %f then,\n", num[2]);
     printf("  %f then,\n", num[3]);
     printf("  %f\n\n", num[4]);

     ans = (num[0] + num[1] + num[2] + num[3] + num[4]) / 5.0;

     printf("The average of these five numbers is: %f\n\n", ans);
     return 0;

}
```

Out put



```
Please enter five real numbers:

  First number: 5.3
  Second number: 9.8
  Third number:  6.4
  Fourth number:  .8
  Fifth number:  98.2


The user enterd:

  5.300000 then,
  9.800000 then,
  6.400000 then,
  0.800000 then,
  98.199997

The average of these five numbers is: 24.100000
```

Start time: 23/march 1:50pm
Finish time: 3:50
Entry: 10

Three tuple maths:

Code

Run

Set up arrays int first[3], second[3], ans[3]

Ask from inputs save into first[0]~[2]

Print “first array is first[0] first[1] first[2]”

Ask for inputs save into second[0]~[2]

Print “second array is second[0] second[1] second[2]”

Print calculating

Ans[0] = first[0] +second[0]
Ans[1] = first [1] + second[1]
Ans[2] = first[2] + second[2]

Print “adding corresponding elements: \{ %d, %d, %d\}, ans[0], ans[1], ans[2];

Ans[0] = first[0] - second[0]
Ans[1] = first[1] – second[1]
Ans[2] = first[2] – second[2]

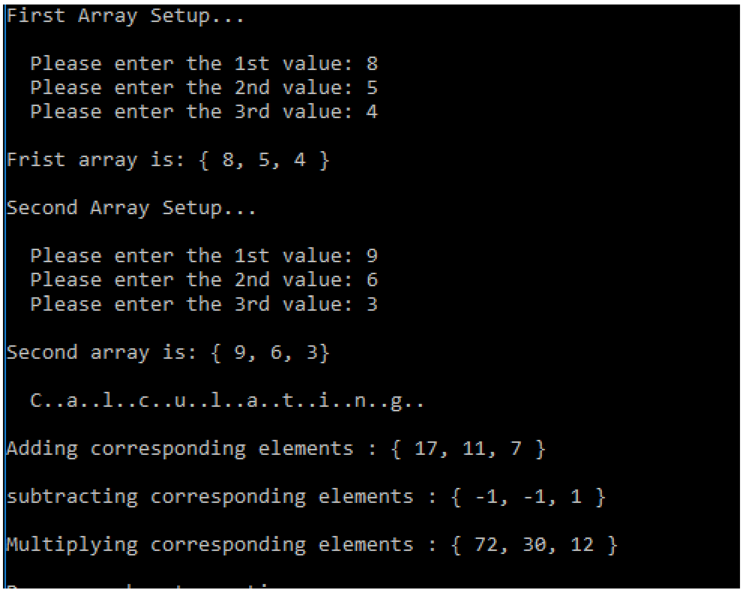
Print ”subtracting corresponding elements: \{ %d, %d, %d\}, ans[0], ans[1], ans[2];

Ans[0] = first[0] * second[0]
Ans[1] = first[1] * second[1]
Ans[2] = first[2] * second[2]

Print “multiplying corresponding elements: \{ %d, %d, %d\}, ans[0], ans[1], ans[2];

```c
#include <stdio.h>

int main()
{
      int first[4];
      int second[4];
      int ans[4];

      printf("First Array Setup...\n\n");
      printf("  Please enter the 1st value: ");
      scanf("%d", &first[0]);
      printf("  Please enter the 2nd value: ");
      scanf("%d", &first[1]);
      printf("  Please enter the 3rd value: ");
      scanf("%d", &first[2]);
```

```
    printf("\n\nFrist array is: \{ %d, %d, %d \}\n\n", first[0], first[1],
first[2]);

    printf("Second Array Setup...\n\n");
    printf("  Please enter the 1st value: ");
    scanf("%d", &second[0]);
    printf("  Please enter the 2nd value: ");
    scanf("%d", &second[1]);
    printf("  Please enter the 3rd value: ");
    scanf("%d", &second[2]);

    printf("\n\nSecond array is: \{ %d, %d, %d\}\n\n", second[0], second[1],
second[2]);

    printf("  C..a..l..c..u..l..a..t..i..n..g..\n\n");

    ans[0] = first[0] + second[0];
    ans[1] = first[1] + second[1];
    ans[2] = first[2] + second[2];

    printf("Adding corresponding elements : \{ %d, %d, %d\}\n\n",
ans[0],ans[1],ans[2]);

    ans[0] = first[0] - second[0];
    ans[1] = first[1] - second[1];
    ans[2] = first[2] - second[2];

    printf("subtracting corresponding elements : \{ %d, %d, %d\}\n\n",
ans[0], ans[1], ans[2]);

    ans[0] = first[0] * second[0];
    ans[1] = first[1] * second[1];
    ans[2] = first[2] * second[2];

    printf("Multiplying corresponding elements : \{ %d, %d, %d\}\n\n",
ans[0], ans[1], ans[2]);

}
```



```
First Array Setup...

  Please enter the 1st value: 8
  Please enter the 2nd value: 5
  Please enter the 3rd value: 4

Frist array is: { 8, 5, 4 }

Second Array Setup...

  Please enter the 1st value: 9
  Please enter the 2nd value: 6
  Please enter the 3rd value: 3

Second array is: { 9, 6, 3}

  C..a..l..c..u..l..a..t..i..n..g..

Adding corresponding elements : { 17, 11, 7 }

subtracting corresponding elements : { -1, -1, 1 }

Multiplying corresponding elements : { 72, 30, 12 }
```

Exe 8

Code

Run

Set up array float e[8]

ask user for real number input save in to array locations e[0]~[7]
scanf(“%8f”,&e);

print “converting float into ascii”

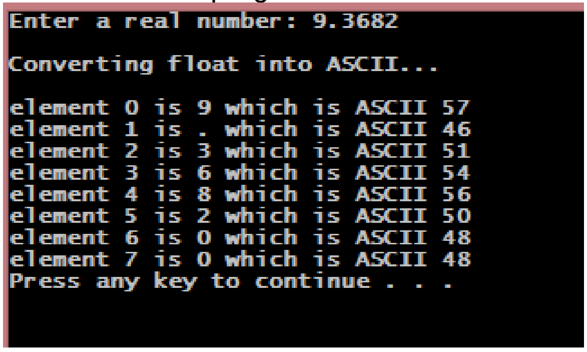
char c[9];

sprintf(c, ”%f’ ,e);

print element 0 is %f which is ascii %d, e, c;

```
#include <stdio.h>
int main()
{

	float e;
	printf("Enter a real number: ");
	scanf("%f", &e);
	printf("\nConverting float into ASCII...\n\n");

	char c[9];
	sprintf(c, "%f", e);
	printf("element 0 is %c which is ASCII %d\n", c[0], c[0]);
	printf("element 1 is %c which is ASCII %d\n", c[1], c[1]);
	printf("element 2 is %c which is ASCII %d\n", c[2], c[2]);
	printf("element 3 is %c which is ASCII %d\n", c[3], c[3]);
	printf("element 4 is %c which is ASCII %d\n", c[4], c[4]);
	printf("element 5 is %c which is ASCII %d\n", c[5], c[5]);
	printf("element 6 is %c which is ASCII %d\n", c[6], c[6]);
	printf("element 7 is %c which is ASCII %d\n", c[7], c[7]);

	return 0;

}
```

The only real problem I had with this code was trying to get my head around the sprintf function it wasn’t until I wrote the program that I understood how it worked.



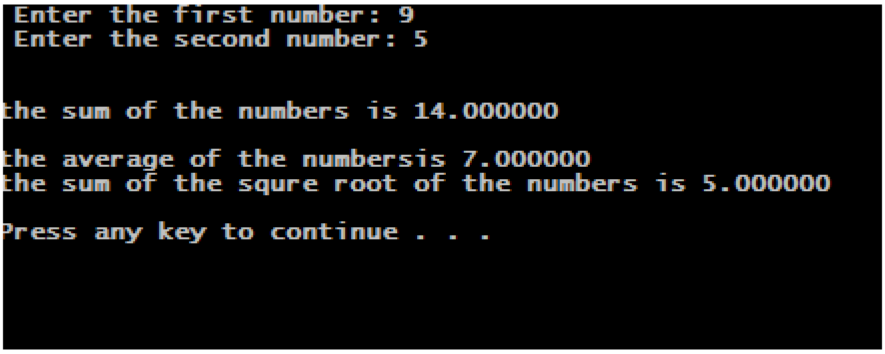
```
Enter a real number: 9.3682

Converting float into ASCII...

element 0 is 9 which is ASCII 57
element 1 is . which is ASCII 46
element 2 is 3 which is ASCII 51
element 3 is 6 which is ASCII 54
element 4 is 8 which is ASCII 56
element 5 is 2 which is ASCII 50
element 6 is 0 which is ASCII 48
element 7 is 0 which is ASCII 48
Press any key to continue . . .
```

Entry 11
Start time:4pm 23 march lab
Finish time: 6 pm

Week4 exe7

These next two programs where done in the practice Lab before the test I decided to put them in because I had to think on then most of the labs in this week where the printf pictures and basic functions we have learned since week 1

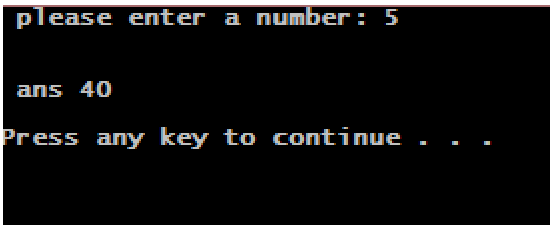
```
#include <stdio.h>
#include <math.h>

int main()

{
        int a;
        int b;
        float ans;

        printf(" Enter the first number: ");
        scanf("%d", &a);
        printf(" Enter the second number: ");
        scanf("%d", &b);

        ans = a + b;
        printf("\n\nthe sum of the numbers is %f\n", ans);

        ans = (a + b) / 2.0;
        printf("\nthe average of the numbersis %f", ans);

        a = sqrt (a);
        b = sqrt(b);
        ans = a + b;
        printf("\nthe sum of the squre root of the numbers is %f\n\n", ans);
        return 0;


}
```



```
 Enter the first number: 9
 Enter the second number: 5


the sum of the numbers is 14.000000

the average of the numbersis 7.000000
the sum of the squre root of the numbers is 5.000000

Press any key to continue . . .
```

Week 4 exe
Quadratic equations

```
#include <stdio.h>

int main()
{
      int a = 1;
      int y = 0;

      printf(" please enter a number: ");
      scanf("%d", &a);

      y = (3 * (a*a)) - (6 * a + 5);
      printf("\n\n ans %d\n\n", y);
      return 0;

}
```

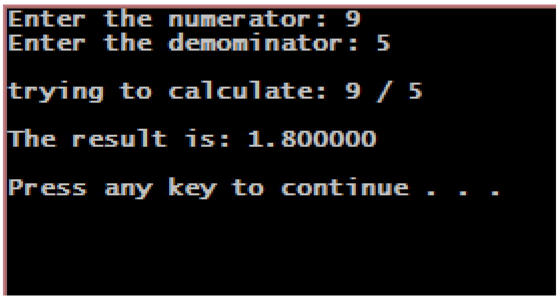


```
 please enter a number: 5


 ans 40

Press any key to continue . . .
```

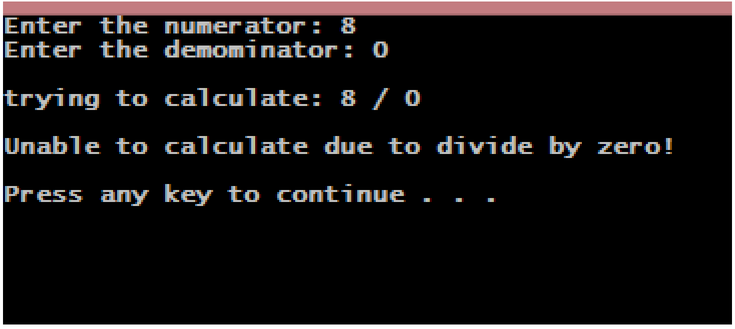
14880673
Entry:12
Time: 4pm 30/3/2016
Finish: 6pm

Week 5

```c
#include <stdio.h>
int main()
{
        float x, y, ans;

        printf("Enter the numerator: ");
        scanf("%f", &x);
        printf("Enter the demominator: ");
        scanf("%f", &y);

        int a, b;
        a = x;
        b = y;

        printf("\ntrying to calculate: %d / %d\n\n", a, b);


        if (y == 0)
        {
                printf("Unable to calculate due to divide by zero!\n\n");
        }
        else
        {
                ans = x / y;
                printf("The result is: %f\n\n", ans);

        }
        return 0;
}
```



```
Enter the numerator: 9
Enter the demominator: 5

trying to calculate: 9 / 5

The result is: 1.800000

Press any key to continue . . .
```

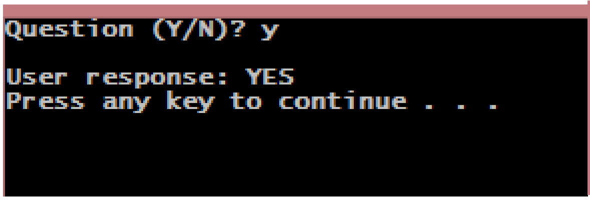


```
Enter the numerator: 8
Enter the demominator: 0

trying to calculate: 8 / 0

Unable to calculate due to divide by zero!

Press any key to continue . . .
```

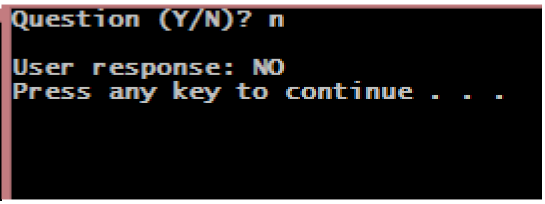
In this lab we get 2 numbers and divide by each other but if the last number entered is a zero the if then statement comes in and says that it cannot be calculated because nothing can be dived by zero others wise it will do the calculations and print them out

Exe 2

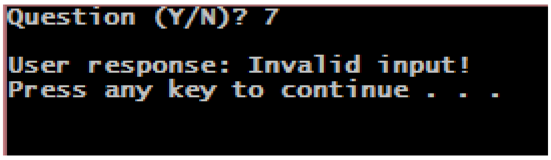
```
#include <stdio.h>
int main()
{
        char input;
        printf("Question (Y/N)? ");

        scanf("%c", &input);

        switch (input)
        {
        case 'Y':
        case 'y':
        {
                printf("\nUser response: YES\n");
                break;
        }
        case 'N':
        case 'n':
        {
                printf("\nUser response: NO\n");
                break;

        }
        default:
        {
                printf("\nUser response: Invalid input!\n");
                break;
        }
        }
}
```



```
Question (Y/N)? y

User response: YES
Press any key to continue . . .
```



```
Question (Y/N)? n

User response: NO
Press any key to continue . . .
```



```
Question (Y/N)? 7

User response: Invalid input!
Press any key to continue . . .
```

In this program we are asking the user to input ‘y’ ‘Y’ ‘n’ or ‘N’ then say what the user inputted be it yes or no but also if the user doesn’t enter a valid input it will say that it was invalid. I found this to be relatively straight forward after re reading the lecture sides and finding out for to do a switch properly. But I did make the mistake of putting a semi colon at the end the variable name ie switch (input);

14880673
Exe 3

```
#include <stdio.h>
int main()

{
    int day;
    printf("Key:    January is 1\n");
    printf("        February is 2\n");
    printf("        March is 3\n");
    printf("        April is 4\n");
    printf("        May is 5\n");
    printf("        June is 6\n");
    printf("        July is 7\n");
    printf("        August is 8\n");
    printf("        September is 9\n");
    printf("        October is 10\n");
    printf("        November is 11\n");
    printf("        December is 12\n");

    printf("\nWhat month were you born in? ");
    scanf("%d", &day);

    if (day < 1 || day > 12)
    {
        printf(" not a vaild month\n\n");
    }
    else if (day == 1)
    {
        printf("\n\nJannuary in New Zealand is in Summer. ");
    }
    else if (day == 2)
    {
        printf("\n\nFeburary in New Zealand is in Summer. ");
    }
    else if (day == 3)
    {
        printf("\n\nMarch in New Zealand is in Autumn. ");
    }
    else if (day == 4)
    {
        printf("\n\nApril in New Zealand is in Autumn. ");
    }
    else if (day == 5)
    {
        printf("\n\nMay in New Zealand is in Autumn. ");
    }
    else if (day == 6)
    {
        printf("\n\nJune in New Zealand is in Winter. ");
    }
    else if (day == 7)
    {
        printf("\n\nJuly in New Zealand is in Winter. ");
    }
    else if (day == 8)
    {
        printf("\n\nAugust in New Zealand is in Winter. ");
    }
    else if (day == 9)
    {
        printf("\n\nSeptember in New Zealand is in Spring. ");
    }
    else if (day == 10)
    {
        printf("\n\nOctober in New Zealand is in Spring. ");
    }
    else if (day == 11)
    {
        printf("\n\nNovember in New Zealand is in Spring. ");
    }
    else if (day == 12)
    {
        printf("\n\nDecember in New Zealand is in Summer.");
    }


    return 0;

}
```

This program was challenging I spent most of the lab time trying to find a easier way to code this without having to repeat and doing a long if else ladder as that was my first choice after talking with the T,A it was discovered that we had not learned how to do that just yet and was the subject of next week's lecture.
I wrote the code as a if else ladder but after I finished I thought what if the user didn't put in a number from 1 to 12 and but it 33 or -3 so I added another if and asked it to find if the input was less than 1 or greater than 12 to say that it was a invalid input

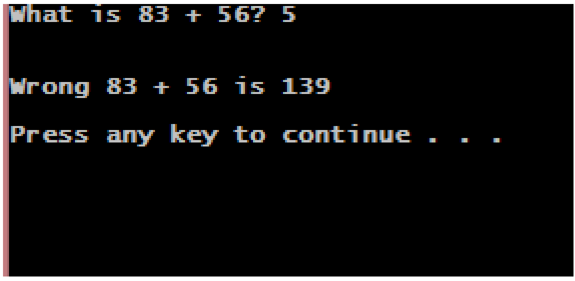
Entry 13
Start Time: 5pm 5/4/16
Finish time: 9pm

Exe 4

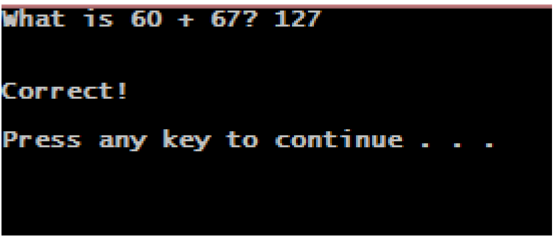
Math quiz

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

int main()
{
    int a, b, c, ans;
    srand(time(0));

    a = (rand() % 100) + 1;
    b = (rand() % 100) + 1;

    ans = a + b;
    printf("What is %d + %d? ", a, b);
    scanf("%d", &c);

    if (ans == c)
    {
        printf("\n\nCorrect!\n\n");
```

```
	}
	else
	{
		printf("\n\nWrong %d + %d is %d\n\n", a, b, ans);
	}

	return 0;

}
```



```
What is 83 + 56? 5


Wrong 83 + 56 is 139

Press any key to continue . . .
```

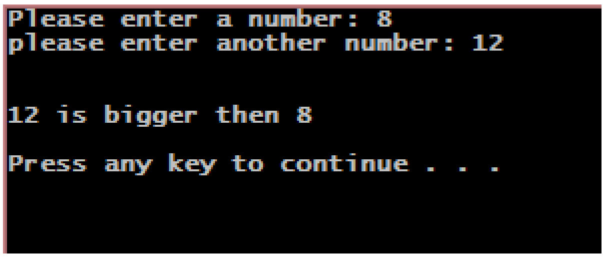


```
What is 60 + 67? 127


Correct!

Press any key to continue . . .
```

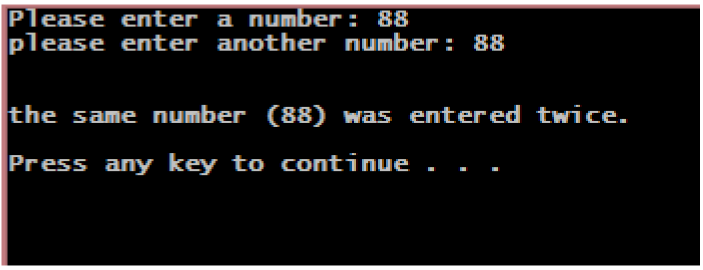
In this program the computer in generating numbers between 1 and 100 and asking the user to add them together if the user gets it right it prints out correct and if it is wrong it says wrong and shows the correct answer

Exe 5

```
#include <stdio.h>

int main()
{
	int a, b;

	printf("Please enter a number: ");
	scanf("%d", &a);
	printf("please enter another number: ");
	scanf("%d", &b);

	if (a == b)
	{
		printf("\n\nthe same number (%d) was entered twice.\n\n", a);
	}
	else if (a < b)
	{
		printf("\n\n%d is bigger then %d\n\n", b, a);
	}
	else if (a > b)
	{
		printf("\n\n%d is bigger then %d\n\n", a, b);
	}

	return 0;
}
```



```
Please enter a number: 8
please enter another number: 12


12 is bigger then 8

Press any key to continue . . .
```



```
Please enter a number: 88
please enter another number: 88


the same number (88) was entered twice.

Press any key to continue . . .
```

In this lab we ask the user to enter 2 numbers the program then checks to see if the same number was entered twice if it was it prints that it was entered twice and shows the number entered if not it compares the numbers and prints out what one is bigger than the other

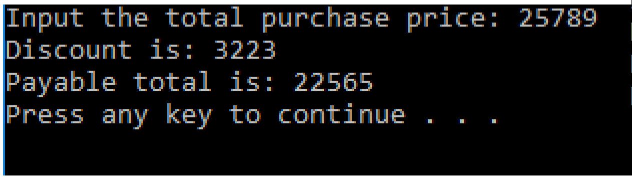
Exe 6

```c
#include<stdio.h>

int main()
{

      int a, b, ans;

      printf("Input the total purchase price: ");
      scanf("%d", &a);

      if (a < 2500)
      {
            printf("Discount is: 0\n");
            printf("Payable total is: %d\n", a);
      }
      else if (a>=2500 && a < 6500)
      {
            b = a * 0.05;
            ans = a - b;
            printf("Discount is: %d\n", b);
            printf("Payable total is: %d\n", ans);
      }
      else if (a >= 6500 && a < 10000)
      {
            b = a * 0.10;
            ans = a - b;
            printf("Discount is: %d\n", b);
            printf("Payable total is: %d\n", ans);
      }
      else if (a >= 10000)
      {
            b = a  *0.125;
            ans = a - b;
            printf("Discount is: %d\n", b);
            printf("Payable total is: %d\n", ans);
      }

      return 0;
}
```

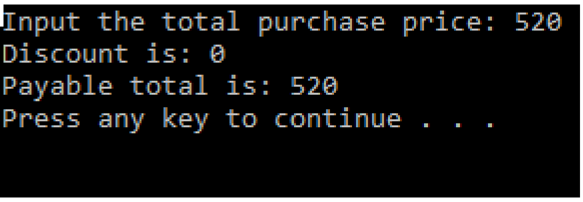
In this program it takes the user input and checks to see if it’s at a sertan value works out a discount and calls back what that is as well as telling you the total after the discount has been applied above is the first attempt at the problem on the face of it it works it says back the number ass needed but another look over the examples in the exercise I noticed that 7777 discount 777 pulled back the number 6999 not 7000 which on the face of it would be correct I then thought my code had to be missing something making the output show 1 number less then would be accurate I thought it must be discount + 1 but that would be an odd thing to do for working out a discount I then thought it must be a reminder so I changed the code take the input as a float and call back as an int this seemed to work. New code below.

14880673

```
#include<stdio.h>

int main()
{

	float a, b, ans;
	int x, y;

	printf("Input the total purchase price: ");
	scanf("%f", &a);

	if (a < 2500.0)
	{
		x = a;
		printf("Discount is: 0\n");
		printf("Payable total is: %d\n", x);
	}
	else if (a >= 2500.0 && a < 6500.0)
	{
		b = a * 0.05;
		ans = a - b;
		x = b;
		y = ans;
		printf("Discount is: %d\n", x);
		printf("Payable total is: %d\n", y);
	}
	else if (a >= 6500.0 && a < 10000.0)
	{
		b = a * 0.10;
		ans = a - b;
		x = b;
		y = ans;
		printf("Discount is: %d\n", x);
		printf("Payable total is: %d\n", y);
	}
	else if (a >= 10000.0)
	{
		b = a  *0.125;
		ans = a - b;
		x = b;
		y = ans;
		printf("Discount is: %d\n", x);
		printf("Payable total is: %d\n", y);
	}

	return 0;
}
```



```
Input the total purchase price: 25789
Discount is: 3223
Payable total is: 22565
Press any key to continue . . .
```

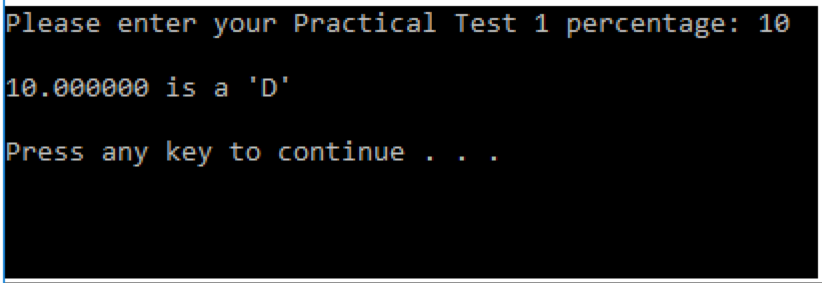


```
Input the total purchase price: 520
Discount is: 0
Payable total is: 520
Press any key to continue . . .
```

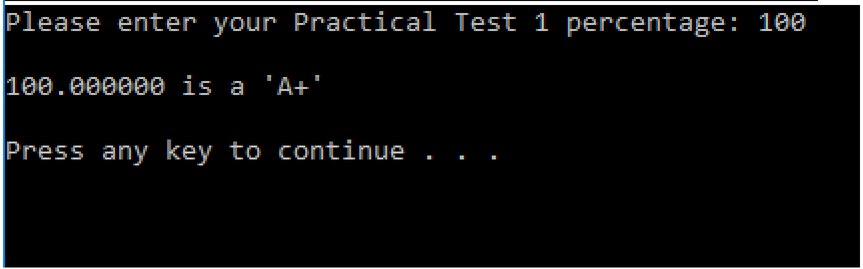
14880673
exe 7
Entry:14
Start time: 1pm 6/4/16
Finish time: 3.30pm

```
#include <stdio.h>

int main()
{
	float mark;

	printf("Please enter your Practical Test 1 percentage: ");
	scanf("%f", &mark);

	if (mark >= 80)
	{
		if (mark >= 90)
		{
			printf("%f is a 'A+'\n\n", mark);
		}
		else if (mark >= 85)
		{
			printf("%f is a 'A'\n\n", mark);
		}
		else if (mark >= 80)
		{
			printf("%f is a 'B'\n\n", mark);
		}
	}
	else if (mark >= 65)
	{
		if (mark >=75)
		{
			printf("%f is a 'B+'\n\n", mark);
		}
		else if (mark >= 70)
		{
			printf("%f is a 'B'\n\n",mark);
		}
		else if ( mark >= 65)
		{
			printf("%f is a 'B-'\n\n", mark);
		}
	}
	else if (mark >= 50)
	{
		if (mark >= 60)
		{
			printf("%f is a 'C+'\n\n", mark);
		}
		else if (mark >= 55)
		{
			printf("%f is a 'C'\n\n", mark);
		}
		else if (mark >= 50)
		{
			printf("%f is a 'C-'\n\n", mark);
		}
	}
	else
	{
		printf("%f is a 'D'\n\n", mark);
	}


}
```

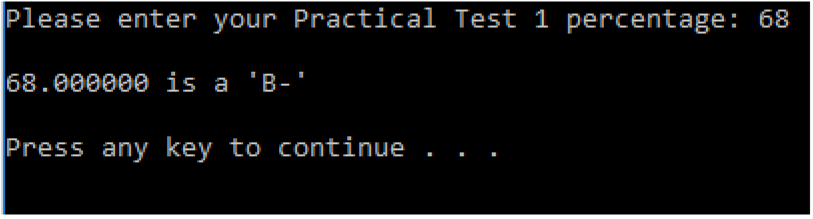
In this exercise its taking the user input as a float and going thought the if statements trying to find the correct line that matched the criteria and doing printing the mark out.



```
Please enter your Practical Test 1 percentage: 10

10.000000 is a 'D'

Press any key to continue . . .
```



```
Please enter your Practical Test 1 percentage: 100

100.000000 is a 'A+'

Press any key to continue . . .
```



```
Please enter your Practical Test 1 percentage: 68

68.000000 is a 'B-'

Press any key to continue . . .
```

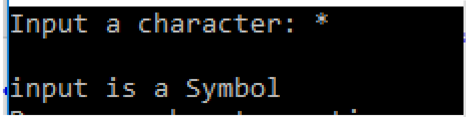
Exe 8

```
int main()

{
	char input;

	printf("Input a character: ");
	scanf("%c", &input);

	if ((input <= 31) || (input >= 127))
	{
		printf("\nInput is non-printable\n");
	}
	else if (input == 32)
	{
		printf("\nInput is Space\n");
	}
	else if ((input >= 48) && (input <= 57))
	{
		printf("\nInput is Digit\n");
	}
	else if ((input >= 65) && (input <= 90))
	{
		printf("\nInput is Uppercase\n");
	}
	else if ((input >= 97) && (input <= 122))
	{
		printf("\nInput is Lowercase\n");
	}
	else
	{
		printf("\ninput is a Symbol\n");
	}
	return 0;
}
```

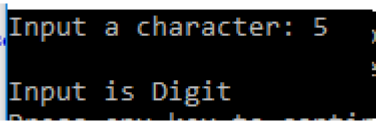
In this code it askes the user for an input off the asci table and checks the if statements to see where it fits and does the instruction in there. The first problem I ran into with this is the if statements I didn't put them in double brackets I only put them in one ie.

If (input <=65) && (input<+122)

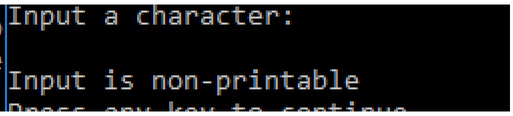
This was showing that it would not work, asked my peers and they said that the if statement was only checking the first not the second and to make it do both you need to bracket them. Output below.



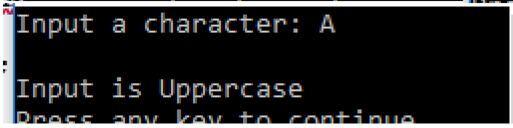
```
Input a character: *

input is a Symbol
```



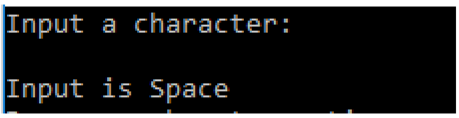
```
Input a character: 5

Input is Digit
```



```
Input a character:

Input is non-printable
```



```
Input a character: A

Input is Uppercase
Press any key to continue
```



```
Input a character:

Input is Space
```

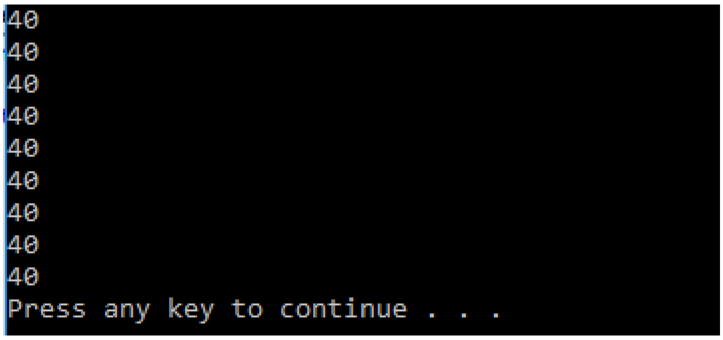
Entry 15
Start Time: 4pm 6/04/16
Finish time: 6pm

Lab tutorial

Week 6 exe 1

```
#include <stdio.h>

int main()
{
	int counter = 9;
	int num = 40;
	while (counter > 0)
	{
		printf("%d\n", num);
		counter--;
		num + 40;

	}
	return 0;
}
```

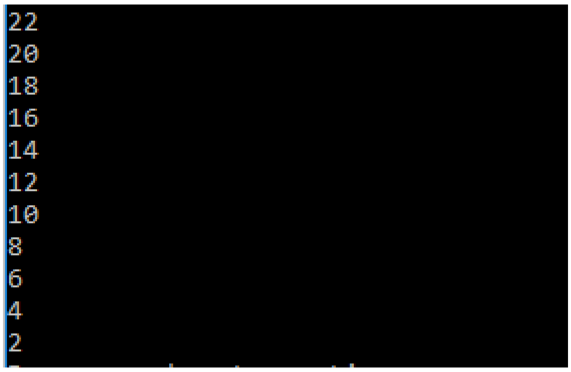
In this program we are looping a counter and printing 40



```
40
40
40
40
40
40
40
40
40
Press any key to continue . . .
```

Exe 2

```
# include<stdio.h>

int main()
{
	int counter = 22;
	while (counter > 0)
	{
		printf("%d\n", counter);
		counter -= 2;
	}
}
```

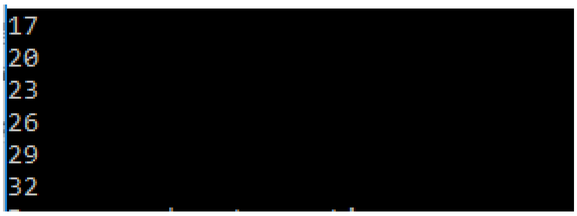


```
22
20
18
16
14
12
10
8
6
4
2
```

In this program we are setting a int value of 22 and setting the loop to print the counter and then take away 2

Exe 3

```
#include <stdio.h>

int main()

{
        int number = 17;
        while (number < 35)
        {
                printf("%d\n", number);
                number += 3;
        }
}
```



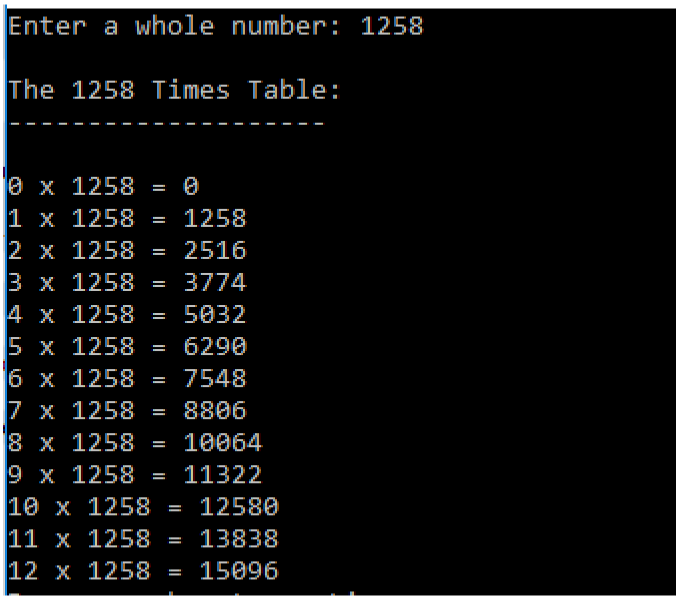
```
17
20
23
26
29
32
```

in this program we are making a loop that adds 3 to the number until it gets greater than 35 then stopes.

Exe 4

```
#include <stdio.h>

int main()

{
        int time = 0;
        int user = 0;
        int ans = 0;

        printf("Enter a whole number:  ");
        scanf("%d", &user);
        printf("\nThe %d Times Table:\n", user);
        printf("--------------------\n\n");

        while (time < 13)
```

```
	{
		printf("%d x %d = %d\n", time, user, ans);
		time++;
		ans = time*user;
	}

}
```



```
Enter a whole number: 1258

The 1258 Times Table:
--------------------

0 x 1258 = 0
1 x 1258 = 1258
2 x 1258 = 2516
3 x 1258 = 3774
4 x 1258 = 5032
5 x 1258 = 6290
6 x 1258 = 7548
7 x 1258 = 8806
8 x 1258 = 10064
9 x 1258 = 11322
10 x 1258 = 12580
11 x 1258 = 13838
12 x 1258 = 15096
```

In this code we are looking at the users inputted times tables. It takes the inputted number and sets up a loop till 12. It takes the user number and multiplies it by counter.

Exe 5

```
#include<stdio.h>

int main()
{
	int start, stop, step, counter;
	counter = 0;

	printf("Starting number ");
	scanf("%d", &start);
	printf("Stopping number ");
	scanf("%d", &stop);
	printf("Step size ");
	scanf("%d", &step);

	printf(" Using a for loop:\n\nStarting at %d...\n", start);

	for (int i = start; i < stop; i += step)
	{
		printf("%d ...\n", i);
		++counter;
	}
	printf("\n\n stoping at %d...", stop);
	printf("\n\nThis loop had %d iterations.", counter);
	return 0;
}
```

In this code we are taking the user inputted numbers for start stop and set then setting up an if loop to count how many time it loops and printing the numbers in the variable

The only problem I can across was trying to find out how to use the if loop statement I first set it up as
If (start; start < stop; start + step) I didn’t know that you had to set up a variable in the if loop I asked the TA and he mentioned that you have to once I did that the program worked fine.

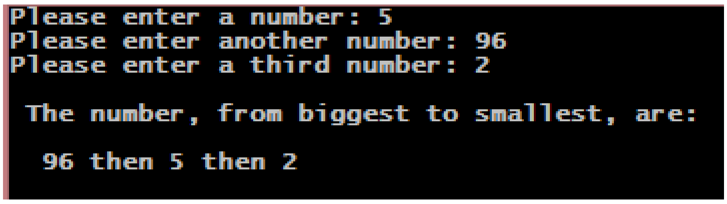
Entry 16
Start Time: 9.30 pm 7/4/16
Finish time: 11.40pm

Exe 9

```
#include <stdio.h>

int main()

{

    int a, b, c;
    int big, sml, mid;

    printf("Please ener a number: ");
    scanf("%d", &a);
    printf("Please enter another number: ");
    scanf("%d", &b);
    printf("Please enter a third number: ");
    scanf("%d", &c);

    if ((a > b) && (a > c))
    {
        big = a;
        if (b > c)
        {
            mid = b;
            sml = c;
        }
        else
        {
            mid = c;
            sml = b;
        }
    }

    else if ((b > a) && (b > c))
    {
        big = b;
        if (a > c)
        {
            mid = a;
            sml = c;
        }
        else
        {
            mid = c;
```

```
                sml = a;

            }
        }
        else if ((c > a) && (c > b))
        {
            big = c;
            if (a > b)
            {
                mid = a;
                sml = b;
            }
            else
            {
                mid = b;
                sml = a;
            }
        }

        printf("\n The number, from biggest to smallest, are:\n\n");
        printf("  %d then %d then %d\n\n", big, mid, sml);
        return 0;


}
```

In this program we are looking at 3 inputted numbers and we a comparing them to find the smallest and largest and the number in the middle it then out put them in order and prints them on the screen.



```
Please enter a number: 5
Please enter another number: 96
Please enter a third number: 2

 The number, from biggest to smallest, are:

  96 then 5 then 2
```

This only problem I came across on this program was trying to get the if statements in the right order.

Exe 10

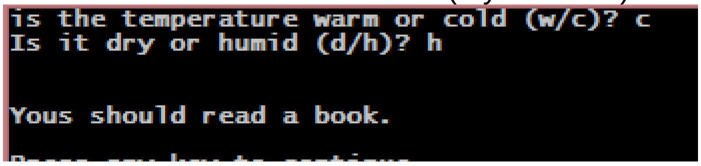
```
#include <stdio.h>

int main()
{
    int a, b, c, total ;

    printf("Enter the first angle ( in degrees ): ");
    scanf("%d", &a);
    printf("Enter the second angle ( in degrees ): ");
    scanf("%d", &b);
    printf("Enther the second angle (in degrees ): ");
    scanf("%d", &c);

    total = a + b + c;
    if (total == 180)
    {
        if ((a == 90) || (b == 90) || (c == 90))
        {
            printf("\n\n %d, %d and %d is a valid right-angle
triangle\n\n",a ,b ,c);
        }
        else
        {
            printf("\n\n %d, %d and %d is a valid triangle\n\n", a, b, c);
        }
    }
    else
    {
        printf("\n\n %d, %d and %d is not a valid triangle.\n\n",a ,b ,c);
    }
}
```

In this code I take the user inputted numbers and checks to see if they add to 180 if they do it then checks if one is 90 degrees if so then says that it's a right angle triangle other ways prints that's it's a triangle.

Exe 11

```
#include <stdio.h>

int main()
{
    char c1, c2;

    printf("is the temperature warm or cold (w/c)? ");
    scanf("%c", &c1);
    printf("Is it dry or humid (d/h)? ");
    scanf(" %c", &c2);

    if (c1 == 'w')
    {
        if (c2 == 'd')
        {
            printf("\n\nYou should play tennis.\n\n");
        }
        if (c2 == 'h')
        {
            printf("\n\nYou should go swimming.\n\n");
        }
    }
    if (c1 == 'c')
    {
        if (c2 == 'd')
        {
            printf("\n\nYou should studdy programming 1.\n\n");
        }
        if (c2 == 'h')
        {
            printf("\n\nYous should read a book.\n\n");
        }
    }
    return 0;
}
```

In this program checks the users input in weather it is w or c (warm or cold) and goes to the correct ladder then checks weather it was d or h (dry or humid) and fallows the instructions in the statement.



```
is the temperature warm or cold (w/c)? c
Is it dry or humid (d/h)? h


Yous should read a book.

Press any key to continue
```

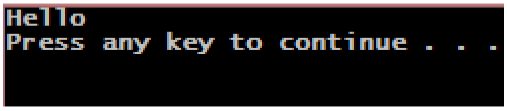
The problem I had with this program was it want picking up the second char or wasn't asking me for it going thought the debugger I still could not see why it was not picking it up then I noticed that it picked up the nul key or enter as its second I put a space before the second %c and that worked it allowed me to put a letter in.

Entry 17:
Time: 4pm 13/4/16
Finish:6 pm

Say hello function

In this program I create a function that prints hello to the screen I then call that function
Only problem I had was getting the call to work I forgot the semi colon and the end

```
#include<stdio.h>

void say_hello()
{
	printf("Hello\n");
}

int main()
{
	say_hello();

	return;
}
```



```
Hello
Press any key to continue . . .
```

Print menu

In the program I create a function that prints a menu set up then I call that function

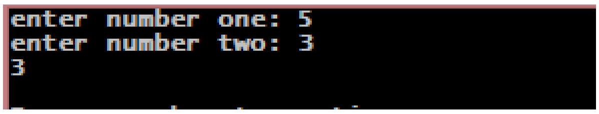
```
#include <stdio.h>

void print_menu();

int main()

{
	print_menu();
}

void print_menu()
{
	printf("Calculator Menu:\n");
	printf("----------------\n\n");
```

```
        printf("1: Add two numbers\n");
        printf("2: Subtract two numbers\n");
        printf("3: Multiply two numbers\n");
        printf("4: Divide two numbers\n");
        printf("5: Quit\n\n");

        return;

}
```

Find minimum

In this program I set up a function to find the minimum entered value the function goes though the if else statements to find the value and the main calls that function and sends the variables out
The main issue I came across with this one was how to get the function to take the variables saved in via the main function I was sending them out after talking with the teacher's assistants they told me I had to name them in the find minimum function.

```
#include<stdio.h>

int find_minimum();


int main()

{
        int p1, p2;
        int min_result;

        printf("enter number one: ");
        scanf("%d", &p1);
        printf("enter number two: ");
        scanf("%d", &p2);

        min_result = find_minimum(p1, p2);

        printf("%d\n\n", min_result);

        return 0;
}

int find_minimum(a, b)

{
        int min_result;
        if (a > b)
        {
                min_result = b;
        }
        else
        {
                min_result = a;
        }

        return min_result;
}
```

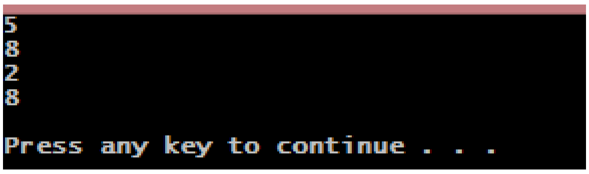


```
enter number one: 5
enter number two: 3
3
```

Find maximum

In this program I make a function that finds the max value of 3 different numbers that the user inputs The main function askes for 3 different numbers then sends them to the find maximum function. The find maximum goes thought some if else statements to find the max number then it sends that back

```
#include<stdio.h>

int find_maximum();

int main()
{
	int p1, p2, p3, max_result;
	scanf("%d", &p1);
	scanf("%d", &p2);
	scanf("%d", &p3);

	max_result = find_maximum(p1, p2, p3);

	printf("%d\n\n", max_result);
	return 0;
}

int find_maximum(a, b, c)
{
	int max_result;
	if ((a > b) && (a > c))
	{
		max_result = a;
	}
	else if ((b > a) && (b > c))
	{
		max_result = b;
	}
	else
	{
		max_result = c;
	}

	return max_result;

}
```

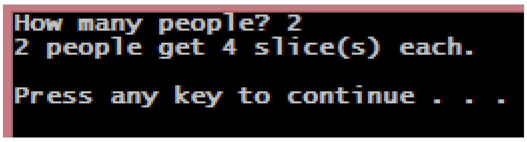


```
5
8
2
8

Press any key to continue . . .
```

Pizza slice program

In this program I create a function that finds the input divided by 8 then sends it back to the main function

```c
#include <stdio.h>

int calculate_pizza_share();

int main()
{
	int number_of_people, slice ;
	printf("How many people? ");
	scanf("%d", &number_of_people);

	slice = calculate_pizza_share(number_of_people);

	printf("%d people get %d slice(s) each.\n\n", number_of_people, slice);

	return 0;

}

int calculate_pizza_share(a)

{
	int slice;

	slice = 8 / a;

	return(slice);
}
```



```
How many people? 2
2 people get 4 slice(s) each.

Press any key to continue . . .
```

Number to be cubed

In the program I create a function that cubes the users input using the math.h
No real issues with these programs as I had learned from the first one how to send the information and use it in the functions.

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

int calculate_cube();

int main()
{
	int num, ans;

	printf("please input number to be cubed: ");
	scanf("%d", &num);
	ans = calculate_cube(num);

	printf("%d cubed is %d\n\n", num, ans);
}

calculate_cube(a)
{
	int ans;
	ans = pow(a, 3);

	return ans;
}
```

get user choice

in this program I create a function that is a void it sends out what the user inputted to the main the main then prints out

```
#include<stdio.h>

int get_user_choice(void)
{
	int choice;

	printf("Enter your choice: ");
	scanf("%d", &choice);

	return choice;
}

int main(choice)

{
	int ans=0;

	ans = get_user_choice();

	printf("%d is your choice", ans);

	return 0;

}
```

Entry 18
Time 9pm 21/04/16
Finish 9:50 pm

Guessing game

14880673
In this program the computer generates a random number then checks to see if the user guessed the correct number though an if else ladder it checks after each input to and if the input was correct it will say what attempt it was on 1st 2nd or 3rd

The part that took the longest on this program was trying to find the best way for the ladder whether it was nested or separate

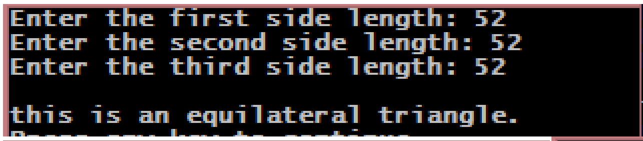
```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main()
{
      srand(time(0));
      int g1, g2, g3, num1;

      num1 = (rand() % 10 ) + 1;

      printf("I'm thinking of a number between 1 and 10...\nYou have three
chances to guess the number...\n\n");
      printf("What is your first guess? ");
      scanf("%d", & g1 );

      if (g1 == num1)
      {
            printf("well done, %d was my number! You took one guess!",g1);
      }
      else
      {
            printf("Inccorect!\n\nWhat is your second guess? ");
            scanf("%d", &g2);
            if (g2 == num1)
            {
                  printf("Well done %d was my number! you took two guesses!",
g2);
            }
            else
            {
                  printf("Incorrect!\n\nWhat is your third guress? ");
                  scanf("%d", & g3 );
                  if (g3 == num1)
                  {
                        printf("Well done %d was my number! it took you three
guesses!",g3);

                  }
                  else
                  {
                        printf("your could not guess my number! it was %d.\n\n",
num1);
                  }
            }
      }

      return 0;
}
```

Entry 19
Time 11pm 22/04/16
Finish 1.50 am 23/04/16

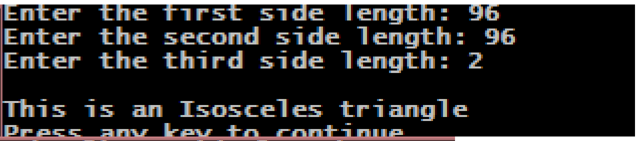
Triangle checker

In this program the computer checks what the user has inputted if they all match it will say that the triangle is an equilateral triangle and if only 2 sides match it will print out isosiles otherwise it’s a scalene triangle There were no real problems I faced in this program.

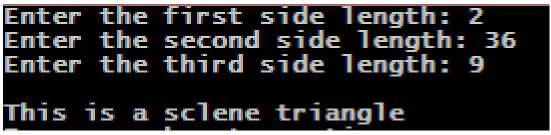
```c
#include <stdio.h>

int  main()
{

      int n1, n2, n3;

      printf("Enter the first side length: ");
      scanf("%d", &n1);
      printf("Enter the second side length: ");
      scanf("%d", &n2);
      printf("Enter the third side length: ");
      scanf("%d", &n3);

      if ((n1 == n2) && (n1 == n3))
      {
            printf("\nthis is an equilateral triangle.\n");
      }
      else if ((n1 == n2) || (n2 == n3) || (n1 == n3))
      {
            printf("\nThis is an Isosceles triangle\n");
      }
      else
      {
            printf("\nThis is a sclene triangle\n");
      }

      return 0;
}
```



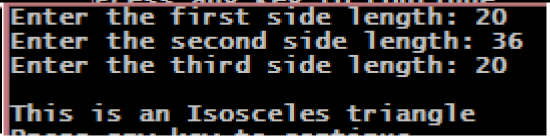
```
Enter the first side length: 52
Enter the second side length: 52
Enter the third side length: 52

this is an equilateral triangle.
Press any key to continue
```



```
Enter the first side length: 96
Enter the second side length: 96
Enter the third side length: 2

This is an Isosceles triangle
Press any key to continue
```



```
Enter the first side length: 2
Enter the second side length: 36
Enter the third side length: 9

This is a sclene triangle
```

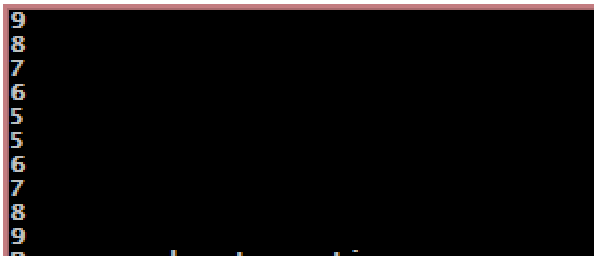


```
Enter the first side length: 20
Enter the second side length: 36
Enter the third side length: 20

This is an Isosceles triangle
Press any key to continue
```

Pseudo Code Loop

In this program it sets up a loop that counts down to 5 then back up from 5 to 9
No real issues arose with this one

```
#include <stdio.h>

int main()
{
	int a,b;

	for (int n = 0; n < 10; ++n)
	{
		if (n > 4)
		{
			printf("%d\n", n);

		}
		else
		{
			a = n;
			b = 9 - a;

			printf("%d\n",b);
		}
	}
}
```

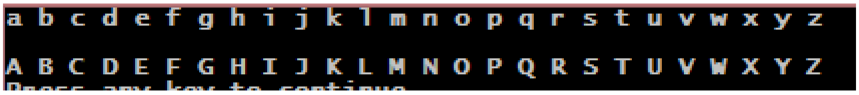


```
9
8
7
6
5
5
6
7
8
9
```

ASCII Alphabet Printer:

In this program I set up 2 for loops, one to print the lower case letters and the other to print the upper case letter
I didn’t really have any problems with this one

```
#include <stdio.h>

int main()
{
	for (int i = 97; i < 123; ++i)
	{
		printf("%c ", i);
	}
	printf("\n\n");
	for (int b = 65; b < 91; ++b)
	{
		printf("%c ", b);
	}

	printf("\n");
	return 0;
}
```



```
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

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
Press any key to continue
```

Lowest and highest input

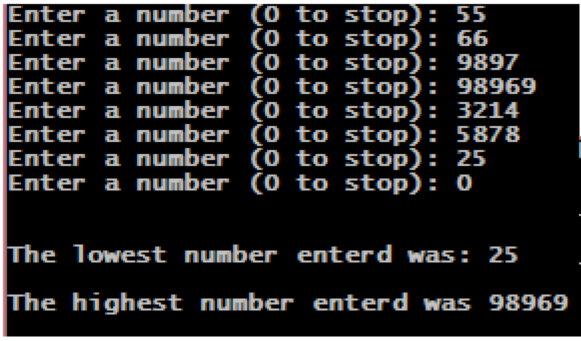
In this program it sets to ints to 0 and 5 then askes the user for and input it then checks to see If that input is higher than 0 and saves it to the new high it also checks to see its lower than 5 and sets it to the new low. It loops back around and asks again does the checks is it higher than the new high if it is then it saves it if not it checks if its lower then low and replaces the low. If the user enters a 0 it will break the loop.
The main problem I had making this program was I set low to 0 so when the program was checking the numbers against low it would not be lower meaning that the lowest would be 0
It took me about 20 min of looking at the code to see this I used the debugger but I didn’t click that that was the problem then I changed it to 5 so it had something to compare

Whilst writing this out I realised that this would not work as what if the user didn’t enter a number lower then 5

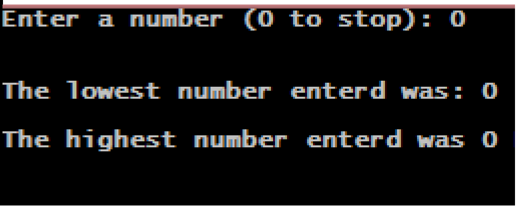
I then changed the low int to be the highest value an int can be meaning any number entered would be lower than it or equal to it
I also added a new if to where if the user entered no numbers and only entered 0 then it would set high and low both to 0

14880673

```c
#include<stdio.h>
int main()

{
	int high = 0, low = 2147483647, counter = 0;

	while (1)
	{
		int input = 0;

		printf("Enter a number (0 to stop): ");
		scanf("%d", &input);

		if (0 == input)
		{
			if (0 == counter)
			{
				high = 0;
				low = 0;
			}
			break;
		}


		if (input > high)
		{
			high = input;
		}
		if ((input < low) && ( input!= 0))
		{
			low = input;
		}
		counter++;

	}
	printf("\n\nThe lowest number enterd was: %d ", low);
	printf("\n\nThe highest number enterd was %d ", high);

	return 0;
}
```



```
Enter a number (0 to stop): 55
Enter a number (0 to stop): 66
Enter a number (0 to stop): 9897
Enter a number (0 to stop): 98969
Enter a number (0 to stop): 3214
Enter a number (0 to stop): 5878
Enter a number (0 to stop): 25
Enter a number (0 to stop): 0


The lowest number enterd was: 25

The highest number enterd was 98969
```



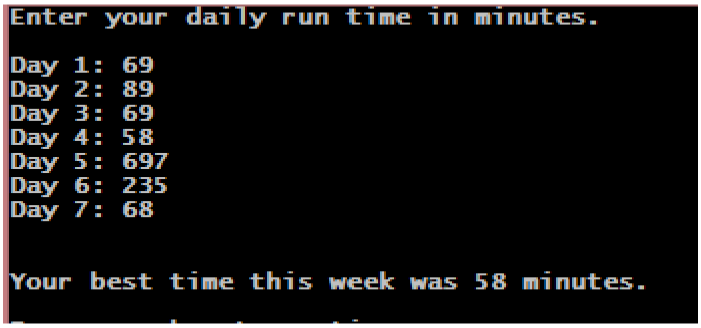
```
Enter a number (0 to stop): 0


The lowest number enterd was: 0

The highest number enterd was 0
```

14880673
Entry: 20
Time 9.30pm 23/04/16
Finish 12am 24/04/16

Best time

In this program the computer is setting up a loop that goes to 7 asks the user to input the time then compares it to the other times and if its less then another time it replaces it.
I didn't really have any problems with this one

```
#include <stdio.h>
int main()
{
    printf("Enter your daily run time in minutes.\n\n");

    int best = 2147483647, input = 0;
    for (int i = 1; i < 8; ++i)
    {
        printf("Day %d: ", i);
        scanf("%d", &input);

        if (input < best)
        {
            best = input;

        }
    }
    printf("\n\nYour best time this week was %d minutes.\n\n",best);

    return 0;
}
```

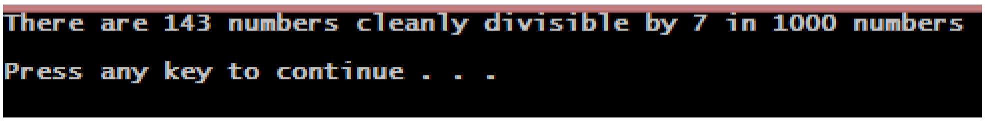


```
Enter your daily run time in minutes.

Day 1: 69
Day 2: 89
Day 3: 69
Day 4: 58
Day 5: 697
Day 6: 235
Day 7: 68


Your best time this week was 58 minutes.
```

How many divisible by seven

This program sets up a for loop to count the how many numbers there are that are cleanly devisable by 7 it adds one to the counter it then displays it
I had no real problems with this program

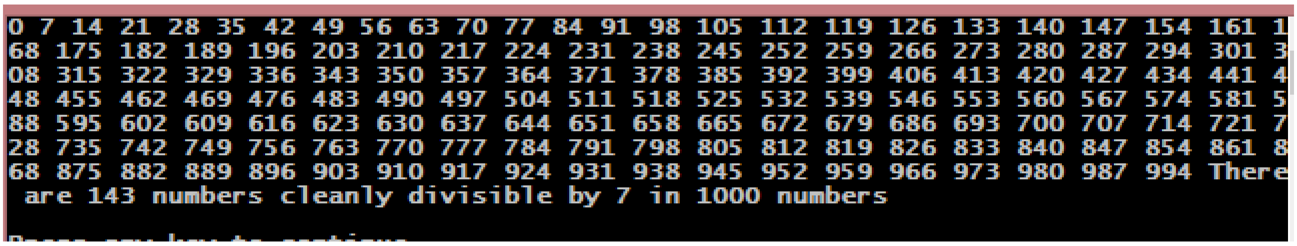
```
#include <stdio.h>

int main()
{
	int counter=0;

	for (int i = 0; i < 1001; ++i)
	{
		if (i % 7 == 0)
		{
			counter++;
		}
	}

	printf("There are %d numbers cleanly devisable by 7 in 1000 numbers\n", 
counter);
	return 0;
}
```



```
There are 143 numbers cleanly divisible by 7 in 1000 numbers
Press any key to continue . . .
```

Whiles making this code I decided to add something to it that would display the numbers that are devisable

```
#include <stdio.h>

int main()
{
	int counter=0;

	for (int i = 0; i < 1001; ++i)
	{
		if (i % 7 == 0)
		{
			printf("%d ",i;
			counter++;
		}
	}

	printf("There are %d numbers cleanly divisible by 7 in 1000 numbers\n", 
counter);
	return 0;
}
```

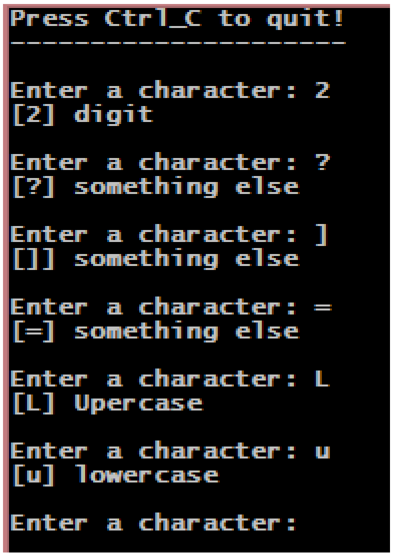


```
0 7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140 147 154 161 1
68 175 182 189 196 203 210 217 224 231 238 245 252 259 266 273 280 287 294 301 3
08 315 322 329 336 343 350 357 364 371 378 385 392 399 406 413 420 427 434 441 4
48 455 462 469 476 483 490 497 504 511 518 525 532 539 546 553 560 567 574 581 5
88 595 602 609 616 623 630 637 644 651 658 665 672 679 686 693 700 707 714 721 7
28 735 742 749 756 763 770 777 784 791 798 805 812 819 826 833 840 847 854 861 8
68 875 882 889 896 903 910 917 924 931 938 945 952 959 966 973 980 987 994 There
 are 143 numbers cleanly divisible by 7 in 1000 numbers

Press any key to continue
```

Infinite Character input loop

In this program the computer set up an infinite loop asking the user to put in a character it then sorts out if it's an upper, lowercase, digit or something else this will go on forever until the user presses control + c the main issue I had with this was when I first compiled and used it would take the [bracket after every input and show that that was "something else"
I changed this by putting a space in front of the %c in the scanf

```
#include <stdio.h>

int main()
{
        char input;
        printf("Press Ctrl_C to quit!\n");
        printf("---------------------\n\n");

        while (1)
        {
                printf("Enter a character: ");
                scanf(" %c", &input);

                if ((input >= 48) && (input <= 57))
                {
                        printf("[%c] digit\n\n", input);
                }
                else if ((input >= 65) && (input <= 90))
                {
                        printf("[%c] Upercase\n\n", input);
                }
                else if ((input >= 97) && (input <= 122))
                {
                        printf("[%c] lowercase\n\n",input);
                }
                else
                {
                        printf("[%c] something else\n\n", input);
                }

        }
}
```



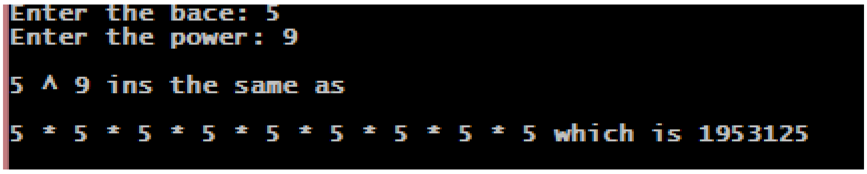
```
Press Ctrl_C to quit!
---------------------

Enter a character: 2
[2] digit

Enter a character: ?
[?] something else

Enter a character: ]
[]] something else

Enter a character: =
[=] something else

Enter a character: L
[L] Upercase

Enter a character: u
[u] lowercase

Enter a character:
```

Entry 21
Time: 8.30pm 24/04/16
Finish:11.30pm

Power loop

This program asks the user for 2 number it then powers the first number input by the second using a for loop
This biggest problem I had with this program was getting the second for loop to display the correct amount of “*” symbols as it would always print one more then needed apon asking a friend he said the you need to look at the second line for you can just add another %d to that and remove one from the loop.

```
#include <stdio.h>

int main()
{
        int bace = 0, pow = 0, ans = 0;

        printf("Enter the bace: ");
        scanf("%d", &bace);
        printf("Enter the power: ");
        scanf("%d", &pow);

        ans = 1;

        for (int i = 1; i <= pow; i++)
        {
                ans = ans * bace;
        }

        printf("\n%d \^ %d ins the same as\n\n", bace, pow);

        for (int c = 1; c < pow; ++c)
        {
                printf("%d * ",bace);

        }

        printf("%d which is %d\n\n",bace, ans);

        return 0;
}
```



```
Enter the bace: 5
Enter the power: 9

5 ^ 9 ins the same as

5 * 5 * 5 * 5 * 5 * 5 * 5 * 5 * 5 which is 1953125
```

Hello input loop

In this program is askes the user for a number that will become the amount of cases it then sets up a for loop with a char array of 100 places it then loops the for the number that the user entered and askes for the test case then prints hello <name>!
It also checks to see if the user has entered a number between 1 and 100 if not it repeats the question
One problem I came across for this one was getting what the user entered to save into the array I then found the %s function
The other is getting the program to check what the user entered is between 1 and 100 I tried using a while loop set to 1 and then using an if statement to with a break in it but it just seems to either stop the whole program or just keep repeating number of test cases spent 2 hours debugging and trying to discover what was going wrong but still could not find the answer.

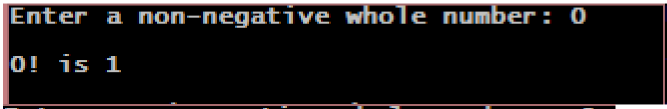
```
#include <stdio.h>

int main()
{
	int cases = 0;

	while (1)
	{
		printf("Number of test cases? ");
		scanf("%d", &cases);

		if ((100 <= cases) && (1 >= cases))
		{
			break;
		}
	}
	for (int i = 1; i < cases; ++i)
	{
		char name[100];
		printf("Test case %d: ",i);
		scanf("%99s", name);
		printf("Output: Hello %s! \n", name);
	}
	return 0;


}
```

14880673
Entry 22
Time: 4pm 25/04/16
Finish: 5.30pm

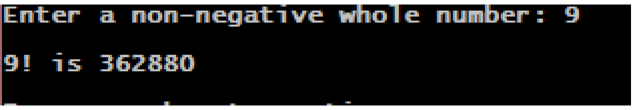
Factorial loop

In this program the user inputs a number if this number is a negative then it will print that the input was bad if the user inputted 0 then it prints 0! Is 1
If the user inputs any other number, then the program will go into the loop and calculate the factorial of that number.
The main problem I had with this program was finding out how to implement the for loop a quick search on google to find out how factorial worked and I clicked.

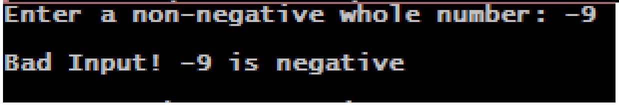
```c
#include <stdio.h>


int main()
{
    int input, fact = 1;

    printf("Enter a non-negative whole number: ");
    scanf("%d", &input);

    if (0 == input)
    {
        printf("\n0! is 1\n\n");
    }
    else if (input < 0)
    {
        printf("\nBad Input! %d is negative\n\n",input);
    }
    else
    {
        for (int i = 1; i <= input; ++i)
        {
            fact = fact*i;
        }
        printf("\n%d! is %d\n\n", input, fact);
    }

}
```



```
Enter a non-negative whole number: 0

0! is 1
```



```
Enter a non-negative whole number: 9

9! is 362880
```



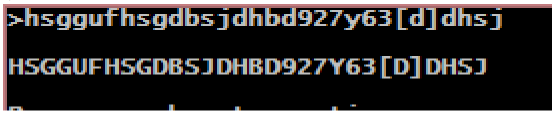
```
Enter a non-negative whole number: -9

Bad Input! -9 is negative
```

14880673

For loop using char array

In this program it takes the user input saves it to a char array it then sets up a for loop and checks to see if it's in the range from 95 ~122 if it is it changes that section of the array to whatever it was – 32 it leaves everything else. It then prints out the new array.
The main problem I had with this one as it kept crashing after the program was done my original program did not have the if input [i] ==0 break after asking friends they suggested that I add a break to stop it repeating over the nulls.

```
#include <stdio.h>

int main()
{
	char input[101];

	printf(">");
	scanf("%100s", input);



	for (char i = 0; i < input; ++i)
	{
		if ((input[i]>=97)&&(input[i]<=122))
		{
			input[i] = input[i] - 32;
		}
		if (input[i] == 0)
		{
			break;
		}
	}

	printf("\n%s", input);

	return 0;
}
```



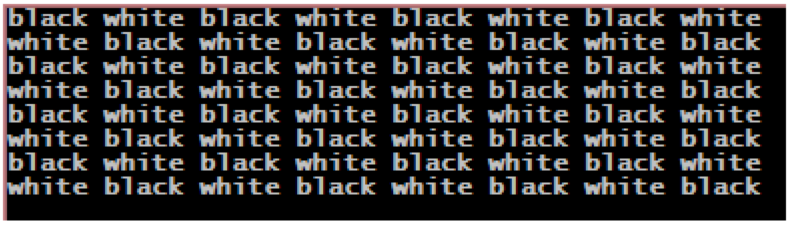
```
>hsggufhsgdbsjdhbd927y63[d]dhsj

HSGGUFHSGDBSJDHBD927Y63[D]DHSJ
```

14880673
Entry: 23
Time: 10 pm 26/04/16
Finish:1 am 27/04/16

Black and white checker board

This program uses for ad nested for loops to print out different lines of black white
I had a lot of trouble with this one not with the syntacs but with actually implementing the different lines I made a code that would print black white on the separate lines but each new line would always start with black not with white in the end I had to call a friend and ask him how he would do this after discussing it we came up with using two different loops one for white black one for black white and just alternating between them

```
#include <stdio.h>

int main()
{
	int rem = 0;

	for (int check = 1; check < 9; ++check)
		{
			rem = check % 2;
			if (rem == 1)
			{
				for (int a = 0; a < 4; ++a)
				{
					printf("black white ");
				}
				printf("\n");
			}
			if(rem == 0)
			{
				for (int b = 0; b < 4; ++b)
				{
					printf("white black ");
				}
				printf("\n");
			}
		}
	printf("\n\n");

	return 0;
}
```

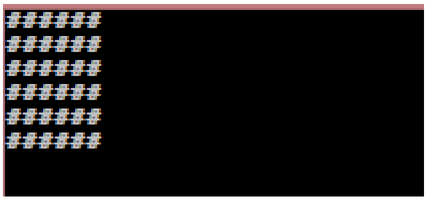


```
black white black white black white black white
white black white black white black white black
black white black white black white black white
white black white black white black white black
black white black white black white black white
white black white black white black white black
black white black white black white black white
white black white black white black white black
```

Nested for loop square

In this program it sets up a for loop that prints out the # and a for loop that prints a new line
I didn’t really have any problems with this program

```c
#include <stdio.h>

int main()
{
	int nl = 0;

	for (nl; nl <= 5; ++nl)
	{
		for (int hash = 0; hash <= 5; ++hash)
		{
			printf("#");
		}
		printf("\n");
	}

	printf("\n\n");


}
```

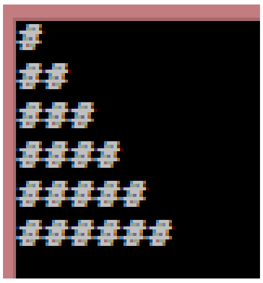


```
######
######
######
######
######
######
```

Right angle triangle

I used the same code as the squre one above just add in an extra int In the program it sets up and int called tri set it to zero it then goes thought the for loops and add one to the tri counter which should add an extra # to the next line
I didn’t have any problems with this as the code is the as the above code for the square

```c
#include <stdio.h>

int main()
{
	int nl = 0;
	int tri = 0;
	for (nl; nl <= 5; ++nl)
	{
		for (int hash = 0; hash <= tri; ++hash)
		{
			printf("#");
		}
		++tri;
		printf("\n");
	}

	printf("\n\n");
}
```

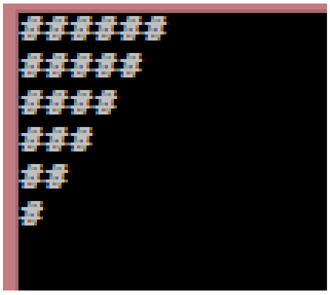


```
#
##
###
####
#####
######
```

Inverted triangle

This program is basically the same as the above code but instead of it adding one # its set up to take it away tri is set to 5 and at the end of the loop its minis 1 from the int of tri
I had no problems with this code as it’s the same are before with e few changes

```
#include <stdio.h>

int main()
{
	int nl = 0;
	int tri = 5;
	for (nl; nl <= 5; ++nl)
	{
		for (int hash = 0; hash <= tri; ++hash)
		{
			printf("#");
		}
		--tri;
		printf("\n");
	}

	printf("\n\n");


}
```

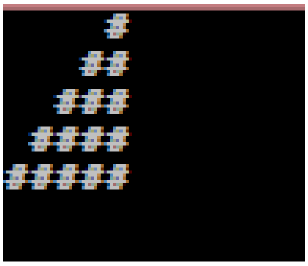


```
######
#####
####
###
##
#
```

14880673
Right aligned triangle

This program is very similar to the right angle triangle before expect I added the changed the new line to space so as to be able to read it eazerly it set up an int called space and scount(space count) sets space to 1 and add to the spaces as the loop goes on it uses the same hash printing code as the others.
This one I had to think about trying to find out how to add in the descending spaces took a trip to google forums

```
#include <stdio.h>

int main()
{
    int nl = 0;
    int tri = 0;
    int numspace = 4;

    for (int space = 1; space <= 5; ++space)
    {
        for (int scount = numspace; scount >= 1; --scount)
        {

        printf(" ");
        }
        for (int hash = 0; hash <= tri; ++hash)
        {
            printf("#");
        }
        ++tri;
        --numspace;
        printf("\n");
    }

    printf("\n\n");
}
```

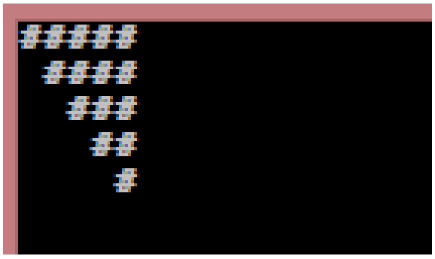


```
    #
   ##
  ###
 ####
#####
```

Inverse triangle right aligned

This program is the same as the one above except ive changed the values of the ints and set them to add 1 is stead of taking away one also I’ve set the number of “#” to add instead of taking away.

```
#include <stdio.h>

int main()
{
        int nl = 0;
        int tri = 4;
        int numspace = 0;

        for (int space = 1; space <= 5; ++space)
        {
                for (int scount = numspace; scount >= 1; --scount)
                {

                printf(" ");
                }
                for (int hash = 0; hash <= tri; ++hash)
                {
                        printf("#");
                }
                --tri;
                ++numspace;
                printf("\n");
        }

        printf("\n\n");
}
```



```
#####
 ####
  ###
   ##
    #
```

Entry 23
Time 10:30pm 29/04/16
Finish:1am 30/04/16

Repeated welcome

Int his program I set up a function that when called prints out welcome to week7 lab the number of times it was called in the main function I added another function the prints out a loop for 27 “-“ to make it look better To real issues with it but I

```
void space(void);

int main()

{

        repeted_welcome(1);
        space();
```

```
    repeted_welcome(4);
    space();
    repeted_welcome(5);
    space();
    repeted_welcome(10);
    space();

    return 0;
}

void repeted_welcome(num)

{
    for (int i = 1; i <= num; ++i)
    {
        printf("Welcome to the Week 7 lab!\n");

    }
    return 0;
}

void space(void)
{
    for (int a = 0; a <= 27; ++a)
    {
        printf("-");
    }
    printf("\n");

}
```

Colour spectrum

In this program I make a function that takes the augment that the main function sends. It goes through an if else ladder to find the correct colour to print.

```
#include <stdio.h>

void print_colour();

int main()
{
    int wave = 0, colour = 0 ;
    printf("Enter a wavelength in nanometres: ");
    scanf("%d", &wave);

    print_colour(wave);
}

void print_colour(wave)

{

    if ((wave <= 700) && (wave >= 635))
    {
        printf("%d is red\n\n",wave);
    }
    else if ((wave <= 634) && (wave >= 590))
    {
        printf("%d is Orange\n\n",wave);
    }
    else if ((wave <= 589) && (wave >= 560))
    {
        printf("%d is yellow\n\n", wave);
    }
    else if ((wave <= 559) && (wave >= 520))
    {
        printf("%d is green\n\n", wave);
    }
    else if ((wave <= 520) && (wave >= 490))
    {
        printf("%d is cyan\n\n", wave);
    }
    else if ((wave <= 489) && (wave >= 450))
    {
        printf("%d is blue\n\n", wave);
    }
    else if ((wave <= 449) && (wave >= 400))
    {
        printf("%d is violet\n\n");
    }
    else
    {
        printf("%d is invisable\n\n", wave);
    }
    return 0;
}
```

Entry:24

Time:4pm 4/5/16

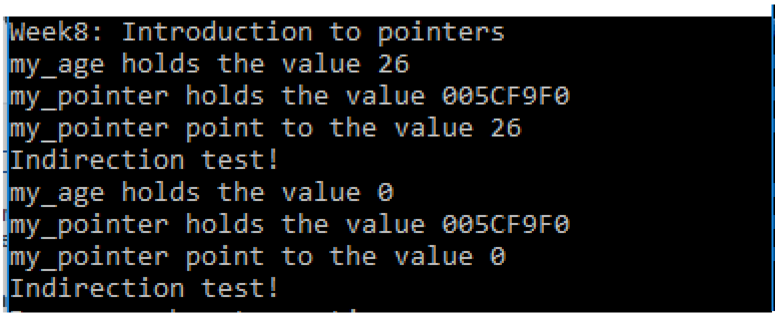
Finish:6pm

In this program I create a int and set it to my age I then crate a pointer to the address of the int that was created. It then prints the variable It then prints he location of the variable. It then prints the information in the variable from the pointer.

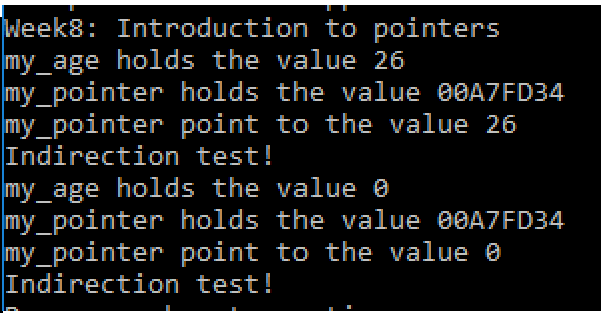
Only issue I came up with was that I did not know how to get the pointer to display the information its pointing to. A quick look on the lecture slides and I found dereferencing and put a * in front and it worked.

```
#include <stdio.h>

int main()
{
    printf("Week8: Introduction to pointers\n");

    int my_age;
    my_age = 26;

    int* my_pointer;
    my_pointer = &my_age;

    printf("my_age holds the value %d\n", my_age);
    printf("my_pointer holds the value %p\n", my_pointer);
    printf("my_pointer point to the value %d\n", *my_pointer);
    printf("Indirection test!\n");
```

```
	*my_pointer = 0;

	printf("my_age holds the value %d\n", my_age);
	printf("my_pointer holds the value %p\n", my_pointer);
	printf("my_pointer point to the value %d\n", *my_pointer);

	return 0;
}
```



```
Week8: Introduction to pointers
my_age holds the value 26
my_pointer holds the value 005CF9F0
my_pointer point to the value 26
Indirection test!
my_age holds the value 0
my_pointer holds the value 005CF9F0
my_pointer point to the value 0
Indirection test!
```

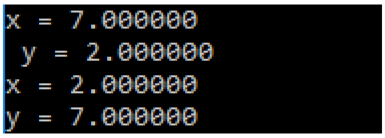


```
Week8: Introduction to pointers
my_age holds the value 26
my_pointer holds the value 00A7FD34
my_pointer point to the value 26
Indirection test!
my_age holds the value 0
my_pointer holds the value 00A7FD34
my_pointer point to the value 0
Indirection test!
```

Float exchange

In this program I create a function that swaps the to numbers in the variables around using pointers, Only issue I came across was I told it to print with %d and not %f.

```
#include <stdio.h>

void exchange();

int main()
{
	float x = 7, y = 2;

	printf("x = %f\n y = %f\n", x, y);

	exchange(&x,&y);

	printf("x = %f\ny = %f\n", x, y);
}
void exchange(float* px,float* py)
{
	float temp = *px;
	*px = *py;
	*py = temp;

	return 0;
}
```



```
x = 7.000000
 y = 2.000000
x = 2.000000
y = 7.000000
```

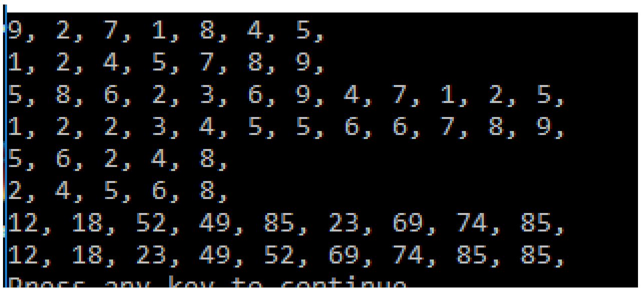
Entry:25
Start 2pm 11/05/16
Finish 6pm

Modularise sort

This program looks at the data sent to it via an array pointer and then sorts it out and prints in on to the screen
One main issue I came across was error c2371 took me a while to figger it out and it was because I put the function after the main function so I needed to put up a prototype
The other issue was trying to find out who to send the information in the int a quick look to the pointers examples in week 8 fixed that one

```
#include <stdio.h>

void print_array();
void sort_array(int* data,int num )
{
    int temp = 0;
    int j = 0;

    for (int i = 1; i < num; ++i)
    {
        temp = data[i];
        j = i - 1;

        while (temp < data[j] && j >= 0)
        {
            data[j + 1] = data[j];
            j = j - 1;
        }

        data[j + 1] = temp;
    }

    return 0;
}

int main()
{
    int data[7] = { 9, 2, 7, 1, 8, 4, 5 };
    int more_data[12] = { 5, 8, 6, 2, 3, 6, 9, 4, 7, 1, 2, 5 };
    int some_more[5] = { 5, 6, 2, 4, 8 };
    int lots_data[9] = { 12, 18, 52, 49, 85, 23, 69, 74, 85 };

    print_array(data, 7);
    sort_array(data, 7);
    print_array(data, 7);

    print_array(more_data, 12);
    sort_array(more_data, 12);
    print_array(more_data, 12);

    print_array(some_more, 5);
    sort_array(some_more, 5);
```

```
        print_array(some_more, 5);

        print_array(lots_data, 9);
        sort_array(lots_data, 9);
        print_array(lots_data, 9);

}

void print_array(int* data, int num)
{
        for (int i = 0; i < num; ++i)
        {
                printf("%d, ", data[i]);
        }

        printf("\n");
}
```



```
9, 2, 7, 1, 8, 4, 5,
1, 2, 4, 5, 7, 8, 9,
5, 8, 6, 2, 3, 6, 9, 4, 7, 1, 2, 5,
1, 2, 2, 3, 4, 5, 5, 6, 6, 7, 8, 9,
5, 6, 2, 4, 8,
2, 4, 5, 6, 8,
12, 18, 52, 49, 85, 23, 69, 74, 85,
12, 18, 23, 49, 52, 69, 74, 85, 85,
Press any key to continue
```

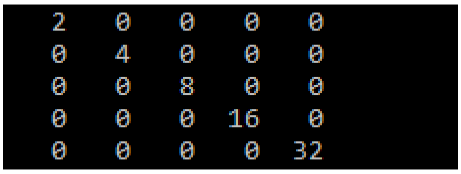
Exe diagonal square array fill

This program loops though the array and then prints out each bit in the array on new lines it also aligns them in spaces of 4
I big issue I had here was on my second int I used ‘o’ which looks a lot like 0 and I used for (int o=0; 0<0; ++o) meaning that I would get an unending loop this mistake was hard to find and took from this I won’t be using ‘o’ for an int again

Next issue was the aligning of the values I could not find a way to align them I tyred to use spaces but that would align in the wrong way asking the TA he said use %4d to make the printf function print values of 4 so if there is 1 character it will print 3 spaces and if there is 2 it would print 2 spaces.

```
#include <stdio.h>

int main()

{
        int five_five[5][5] =
        {
                { 2, 0, 0, 0, 0 },
                { 0, 4, 0, 0, 0 },
                { 0, 0, 8, 0, 0 },
                { 0, 0, 0, 16, 0 },
                { 0, 0, 0, 0, 32 }
        };

        for (int i = 0; i < 5; ++i)
        {
                for (int o = 0; o < 5; ++o)
                {
```

```
            printf("%*d",4, five_five[i][o]);
        }
        printf("\n");
    }

    return 0;
}
```



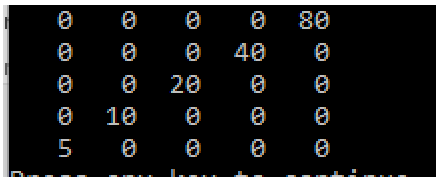
```
   2   0   0   0   0
   0   4   0   0   0
   0   0   8   0   0
   0   0   0  16   0
   0   0   0   0  32
```

upon re reading the question I think I was meant to set up the array like this

```
#include <stdio.h>

int main()

{
    int x = 0;
    int y = 0;
    int counter = 2;
    int five_five[5][5] =
    {
        { 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 0 }
    };

    while (counter <= 32)
    {
        five_five[x][y] = counter;
        ++x;
        ++y;
        counter = counter * 2;
    }

    for (int i = 0; i < 5; ++i)
    {
        for (int o = 0; o < 5; ++o)
        {
            printf("%*d",4, five_five[i][o]);
        }
        printf("\n");
    }

    return 0;
}

#include <stdio.h>

int main()

{
    int x = 0;
```

```
	int y = 4;
	int counter = 80;
	int five_five[5][5] =
	{
		{ 0, 0, 0, 0, 0 },
		{ 0, 0, 0, 0, 0 },
		{ 0, 0, 0, 0, 0 },
		{ 0, 0, 0, 0, 0 },
		{ 0, 0, 0, 0, 0 }
	};

	while (counter >= 5)
	{
		five_five[x][y] = counter;
		++x;
		--y;
		counter = counter / 2;
	}

	for (int i = 0; i < 5; ++i)
	{
		for (int o = 0; o < 5; ++o)
		{
			printf("%*d", 4, five_five[i][o]);
		}
		printf("\n");
	}

	return 0;
}
```

Exe diagonal square array fill again
In this program it is essentially the same as above but its got the numbers going in the other way
The main problem I had in this is the off by one error with both myself and the ta looking for a while he found that because I set the y value to 5 and not 4 everything would be out of alignment



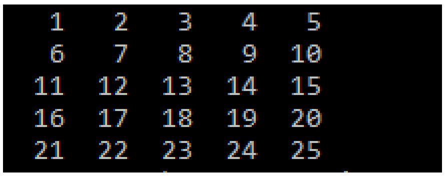
```
   0   0   0   0  80
   0   0   0  40   0
   0   0  20   0   0
   0  10   0   0   0
   5   0   0   0   0
```

Exe square array fill

This program adds numbers to the array whilst filling it up too from 1 to 25 it then prints it out

The issues I had with this one is in my first for loop it set it up to do it 25 times which made it go way to far in the code took me a while to find out that it was 5x5 so the first for loop needed to be less then or equal to 4

```
#include <stdio.h>

int main()
{
	int five_five[5][5];
	int counter = 1;
	for (int i = 0; i <= 4; ++i)
	{
		for (int f = 0; f <= 4; ++f)
		{
			five_five[i][f] = counter;
			counter++;
			printf("%4d",five_five[i][f]);
		}

		printf("\n");
	}
}
```



```
   1   2   3   4   5
   6   7   8   9  10
  11  12  13  14  15
  16  17  18  19  20
  21  22  23  24  25
```

Entry 26
Start 4pm 18 /05/16
Finish 6pm

Week 10
The cat structure

```
#include <stdio.h>

struct cat
{
	char name[50];
	int age;
	float weight;
	float tail;
};

int main()
{
	struct cat wiskers;

	sprintf(wiskers.name, "wiskers");
	wiskers.age = 7;
	wiskers.tail = 15;
	wiskers.weight = 8;

	printf("A cat..\n\n");
	printf("Name:         %s\n", wiskers.name);
	printf("Age:          %d\n", wiskers.age);
	printf("Weight:       %f\n", wiskers.weight);
	printf("Tail length   %f\n", wiskers.tail);

	return 0;
}
```

14880673
This program creates a structure called cat in that structure it has a name and age as well as weight and tail I then put in the correct values for the cat named whiskers and print those on to the screen
No real problems me making the code just a look over the examples codes

```
Sizeof
#include <stdio.h>

struct person
{
	char first_inital;
	char last_inital;
	int age;
	int weight;
	char sex;
	int height;
	char blood_type;
	int shoe_type;
};

struct optimisedperson
{
	int shoe_type;
	int height;
	int age;
	int weight;
	char first_inital;
	char last_inital;
	char blood_type;

};

int main()

{
	printf("size? %d bytes.\n", sizeof(struct person));
	printf("size? %d bytes.\n", sizeof(struct optimisedperson));
}
```

In this program a structure is setup with different variables in an optimised order and another is set up with them in the correct order or optimised order the program then prints the size of both the structures showing the size of the min bytes
Only problem I came across was how to use the size of initially I did not use the struct I had print sizeof(person) not sizeof(struct person). A quick look on the slides ad this was found.

Exe 3

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


float compute_distance3d(struct Point3D first, struct Point3D end);
struct Point3D
{
	float x;
	float y;
	float z;
};

int main()
```

```
{
        float dist;
        float p1x, p1y, p1z;
        float p2x, p2y, p2z;


        printf("enter your first point coordanates (3) ");
        scanf("%f", &p1x);
        scanf("%f", &p1y);
        scanf("%f", &p1z);

        printf("enter you sencond point coordanates (3) ");
        scanf("%f", &p2x);
        scanf("%f", &p2y);
        scanf("%f", &p2z);

        struct Point3D start;

        start.x = 1.0;
        start.y = 2.0;
        start.z = 3.0;

        struct Point3D finish;

        finish.x = 4.0;
        finish.y = 5.0;
        finish.z = 6.0;

        dist = compute_distance3d(start, finish);

        printf("%f", dist);
}

float compute_distance3d(struct Point3D first, struct Point3D end)
{
        float distance = sqrt(pow((first.x - end.x), 2) + pow((first.y - end.y),
2) + pow((first.z - end.z), 2));

        return distance;
}
```

Entry 27
Time 5 pm 23/05/16
Finish 11pm

Pass by reference with pointers

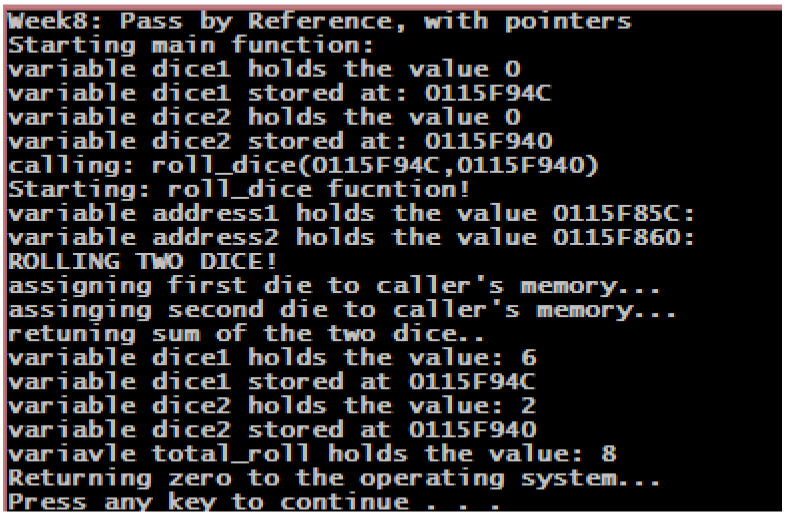
In this program we set up a dice roll function that rolls some random numbers and saves them in the variable in the main function using pointers. The main function sets up the variables to save the dice rolls to and sends the address of them to the dice roll function the dice roll function also returns the sum of the 2 dice rolls

The biggest problem I had in this one was saving the dice rolls in to the variables in the main function.
In my first attempt I had the points set up like [address1 = (rand()%6)+1] and I was looking everywhere to find where this might be going wrong because the code kept crashing on the line where I add the two

pointers together when thought the debugger and all that to see if I could see anything different. I found that the dice rolls were not being saved and it kept saying that they were 0 again I did not click to the dereferencing error I had. Walked away came back a little latter had another look at the sides and the examples saw that I had forgotten to put the * in front of the address1 recompiled and it worked
Code below

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

int roll_dice(int* address1, int* address2)
{
      int total;

      printf("Starting: roll_dice fucntion!\n");
      printf("variable address1 holds the value %p:\n", &address1);
      printf("variable address2 holds the value %p:\n", &address2);
      printf("ROLLING TWO DICE!\n");

      *address1 = (rand() % 6) + 1;

      printf("assigning first die to caller's memory... \n");

      *address2 = (rand() % 6) + 1;

      printf("assinging second die to caller's memory... \n");

      total =  *address1 + *address2;

      printf("retuning sum of the two dice..\n");

      return total;

}
int main()
{
      srand(time(0));

      int dice1 = 0, dice2 = 0;
      int total_roll;

      printf("Week8: Pass by Reference, with pointers\n");
      printf("Starting main function:\n");
      printf("variable dice1 holds the value %d\n", dice1);
      printf("variable dice1 stored at: %p\n", &dice1);
      printf("variable dice2 holds the value %d\n", dice2);
      printf("variable dice2 stored at: %p\n", &dice2);
      printf("calling: roll_dice(%p,%p)\n", &dice1, &dice2);

      total_roll = roll_dice(&dice1, &dice2);

      printf("variable dice1 holds the value: %d\n", dice1);
      printf("variable dice1 stored at %p\n", &dice1);
      printf("variable dice2 holds the value: %d\n", dice2);
      printf("variable dice2 stored at %p\n", &dice2);
      printf("variavle total_roll holds the value: %d\n", total_roll);
      printf("Returning zero to the operating system...\n");
```

```
        return 0;

}
```



```
Week8: Pass by Reference, with pointers
Starting main function:
variable dice1 holds the value 0
variable dice1 stored at: 0115F94C
variable dice2 holds the value 0
variable dice2 stored at: 0115F940
calling: roll_dice(0115F94C,0115F940)
Starting: roll_dice fucntion!
variable address1 holds the value 0115F85C:
variable address2 holds the value 0115F860:
ROLLING TWO DICE!
assigning first die to caller's memory...
assinging second die to caller's memory...
retuning sum of the two dice..
variable dice1 holds the value: 6
variable dice1 stored at 0115F94C
variable dice2 holds the value: 2
variable dice2 stored at 0115F940
variavle total_roll holds the value: 8
Returning zero to the operating system...
Press any key to continue . . .
```

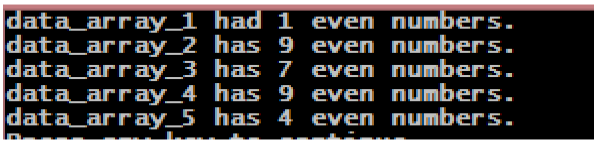
exe passing arrays to functions via pointer:

in this program we set up a function that goes thought the array counting finding all the even numbers and adding one to the counter if the numbers is even it then returns that number.
In the main function we set up an array of different sizes and numbers and send that though to the count_even function.

The main problem I had with this program was trying to find out how to go through the array and get the needed number took me a while but looking thought the previous exercises I found how to do it.

```
#include < stdio.h>

int count_even(int* data_array, int size)
{
        int counter = 0;

        for (int i = 0; i <= size; ++i)
        {

                if (data_array[i]%2 == 0)
                {
                        counter++;
                }
        }

        return counter;
}


int main()
{
        int data_array_1[] = { 1, 3, 5, 7, 9, 11 };
        int data_array_2[] = { 2, -4, 6, -8, 10, -12 ,14, -16 };
        int data_array_3[] = { 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 };
        int data_array_4[] = { 14, 2, 5, 6, 3, 7, 89, 4, 6, 78, 0, 4 };
        int data_array_5[] = { 56, 78, 90, 45, 37, 345, 7 };
```

```
    int result_1 = count_even(data_array_1, 6);

    printf("data_array_1 had %d even numbers.\n", result_1);

    int result_2 = count_even(data_array_2, 8);

    printf("data_array_2 has %d even numbers.\n", result_2);

    int result_3 = count_even(data_array_3, 11);

    printf("data_array_3 has %d even numbers.\n", result_3);

    int result_4 = count_even(data_array_4, 12);

    printf("data_array_4 has %d even numbers.\n", result_4);

    int result_5 = count_even(data_array_5, 7);

    printf("data_array_5 has %d even numbers.\n", result_5);


    return 0;

}
```



```
data_array_1 had 1 even numbers.
data_array_2 has 9 even numbers.
data_array_3 has 7 even numbers.
data_array_4 has 9 even numbers.
data_array_5 has 4 even numbers.
```

Entry 27
Time 7.30 pm 24/05/16
Finish 9pm

Memory sharing with pointer

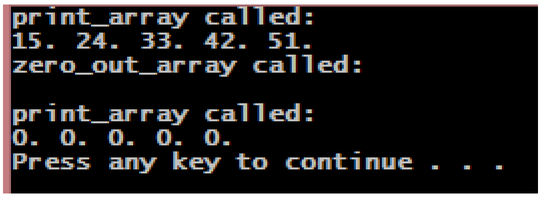
In this program I set up 2 functions one to print one to set each element to 0 in the array using the pointers to access the array

I had a lot of trouble with this one trying to find out how to access the elements in the array via the pointers and how to set them to 0 after I had accessed them after talking with a friend and him giving me pointers in to how they work I found a way to accces the elements but them when I ran the program the program would finish and crash with a run time error #2 array corrupted after I while I round that I was saving another 0 on the array giving me 6 elements once I fixed that all working well

Code below

14880673

```
#include <stdio.h>

void zero_out_array(int* p_array, int num_elements)
{
	printf("zero_out_array called:\n");


	for (int y = 0; y < num_elements; ++y)
	{
		*p_array++ = 0;
	}
	printf("\n");
}

void print_array(int* P_array, int num_elements)
{
	printf("print_array called:\n");

	for (int k = 0; k < num_elements; ++k)
	{
		printf("%d. ", P_array[k]);
	}
	printf("\n");
}

int main()
{
	int main_array[] = { 15, 24, 33, 42, 51 };

	print_array(main_array, 5 );
	zero_out_array(main_array, 5);
	print_array(main_array, 5);

	return 0;
}
```



```
print_array called:
15. 24. 33. 42. 51.
zero_out_array called:

print_array called:
0. 0. 0. 0. 0.
Press any key to continue . . .
```

Entry 28
Time 10 pm 24/05/16
Finish 11.30

Pass by reference, multiple return values:

This program need to count the amount of upper and lower case letters as well as numbers then return and print that value out

I cannot find out how to do this I have spent an hour and a half looking thought lecture sides and code examples as well as using the internet still cannot see anything that would help

```
#include <stdio.h>

void count_categores(char* p_cstring, int* p_upper_count, int* p_lower_count,
int* p_digit_count)
{


}

int main()
{
     char buffer1[] = "hello programing 1 students:";
     char buffer2[] = "learn to program using arrays and pointers!";

     int upper_count = 0;
     int lower_count = 0;
     int digit_count = 0;

     count_catergories(buffer1, &upper_count, &lower_count, &digit_count);

     printf("[%s] upper: %d, lower: %d, digit: %d\n", buffer1, upper_count,
lower_count, digit_count);

     count_categores(buffer2, &upper_count, &lower_count, &digit_count);

     prinft("[%s] upper: %d, lower: %d, digit: %d\n",buffer2, upper_count,
lower_count, digit_count, );

     return 0;
}
```

Entry 29
Time 12.45pm 25/05/2016
Finish 3.45 pm

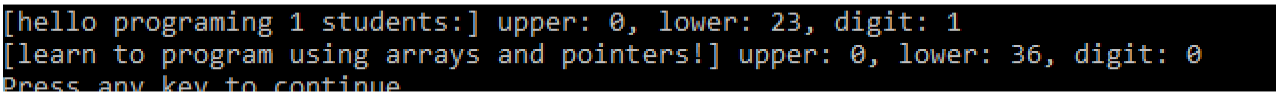
Pass by reference, multiple return values:

This program need to count the amount of upper and lower case letters as well as numbers then return and print that value out

Apon revisiting this question from yesterday I found that I needed to iterate thought the c sting to find the numbers and upper and lowercase via if statements

14880673
the code became

```
#include <stdio.h>

void count_catergories(char* p_cstring, int* p_upper_count, int* p_lower_count,
int* p_digit_count)
{
        *p_digit_count = 0;
        *p_upper_count = 0;
        *p_lower_count = 0;

        for (int y = 0; p_cstring[y] != 0; ++y)
        {
                if ((p_cstring[y] >= 48) && (p_cstring[y] <= 57))
                {
                        *p_digit_count+=1;
                }
                if ((p_cstring[y] >= 65) && (p_cstring[y] <= 90))
                {
                        *p_upper_count += 1;
                }
                if ((p_cstring[y] >= 97) && (p_cstring[y] <= 122))
                {
                        *p_lower_count += 1;
                }
        }

        return 0;

}

int main()
{
        char buffer1[] = "hello programing 1 students:";
        char buffer2[] = "learn to program using arrays and pointers!";

        int upper_count = 0;
        int lower_count = 0;
        int digit_count = 0;

        count_catergories(buffer1, &upper_count, &lower_count, &digit_count);

        printf("[%s] upper: %d, lower: %d, digit: %d\n", buffer1, upper_count,
lower_count, digit_count);

        count_catergories(buffer2, &upper_count, &lower_count, &digit_count);

        printf("[%s] upper: %d, lower: %d, digit: %d\n", buffer2, upper_count,
lower_count, digit_count);

        return 0;
}
```



```
[hello programing 1 students:] upper: 0, lower: 23, digit: 1
[learn to program using arrays and pointers!] upper: 0, lower: 36, digit: 0
Press any key to continue
```

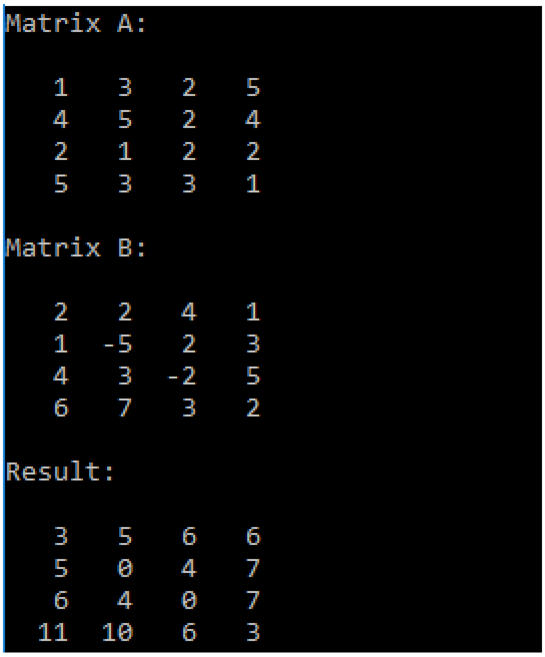
Matrix addition

In this program I set up 2 2d arrays and full them with numbers I then create a 2d array that adds the two numbers in the other arrays together and prints them out on the screen

I didn’t really have any problems in this program

```c
#include <stdio.h>

int main()
{
    int matrix_a[4][4] =
    {
        { 1, 3, 2, 5 },
        { 4, 5, 2, 4 },
        { 2, 1, 2, 2 },
        { 5, 3, 3, 1 },
    };

    int matrix_b[4][4] =
    {
        { 2, 2, 4, 1 },
        { 1, -5, 2, 3 },
        { 4, 3, -2, 5 },
        { 6, 7, 3, 2 },
    };

    int result[4][4];

    printf("Matrix A:\n\n");

    for (int y = 0; y < 4; ++y)
    {
        for (int u = 0; u < 4; ++u)
        {
            printf("%4d", matrix_a[y][u]);
        }

        printf("\n");
    }

    printf("\nMatrix B:\n\n");

    for (int q = 0; q < 4; ++q)
    {
        for (int w = 0; w < 4; ++w)
        {
            printf("%4d", matrix_b[q][w]);
        }

        printf("\n");
    }

    printf("\nResult:\n\n");

    for (int g = 0; g < 4; ++g)
```

```
	{
		for (int h = 0; h < 4; ++h)
		{
			result[g][h] = matrix_a[g][h] + matrix_b[g][h];

			printf("%4d", result[g][h]);
		}

		printf("\n");
	}

	return 0;

}
```



```
Matrix A:

   1   3   2   5
   4   5   2   4
   2   1   2   2
   5   3   3   1

Matrix B:

   2   2   4   1
   1  -5   2   3
   4   3  -2   5
   6   7   3   2

Result:

   3   5   6   6
   5   0   4   7
   6   4   0   7
  11  10   6   3
```

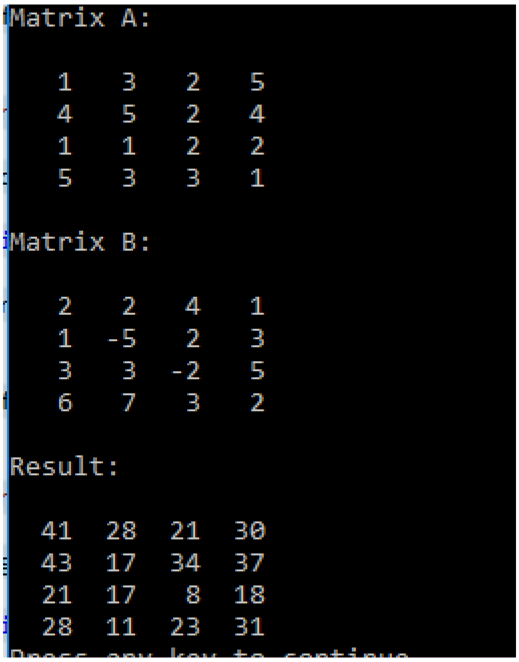
Matrix multiplication

In this program 2 2d arrays are set up and then multiplied together

Intentionally I set it up same as the program above put changed the ‘+’ to a ‘*’ this only multiplied the elements on the 2d array together rather than multiplying them together I had to look on stack overflow to find the answer to this as I could not figure it out I got the answer from [4]

Code below

```
#include <stdio.h>

int main()
{
	int matrix_a[4][4] =
	{
		{ 1, 3, 2, 5 },
		{ 4, 5, 2, 4 },
		{ 1, 1, 2, 2 },
		{ 5, 3, 3, 1 },
	};

	int matrix_b[4][4] =
	{
        { 2, 2, 4, 1 },
        { 1, -5, 2, 3 },
        { 3, 3, -2, 5 },
        { 6, 7, 3, 2 },
    };

    int result[4][4] = { 0 };

    printf("Matrix A:\n\n");

    for (int y = 0; y < 4; ++y)
    {
        for (int u = 0; u < 4; ++u)
        {
            printf("%4d", matrix_a[y][u]);
        }

        printf("\n");
    }

    printf("\nMatrix B:\n\n");

    for (int q = 0; q < 4; ++q)
    {
        for (int w = 0; w < 4; ++w)
        {
            printf("%4d", matrix_b[q][w]);
        }

        printf("\n");
    }

    printf("\nResult:\n\n");

    for (int g = 0; g < 4; ++g)
    {
        for (int h = 0; h < 4; ++h)
        {
            for (int n = 0; n < 4; ++n)
            {
                result[g][h] += matrix_a[g][n] * matrix_b[n][h];
            }

            printf("%4d", result[g][h]);
        }

        printf("\n");
    }

    return 0;

}
```



```
Matrix A:

   1   3   2   5
   4   5   2   4
   1   1   2   2
   5   3   3   1

Matrix B:

   2   2   4   1
   1  -5   2   3
   3   3  -2   5
   6   7   3   2

Result:

  41  28  21  30
  43  17  34  37
  21  17   8  18
  28  11  23  31
Press any key to continue
```

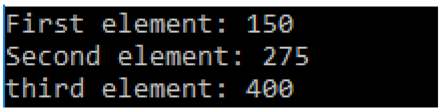
Small malloc

In this program I set up a int to go on the heap then it checks to make sure its not pointing to nothing
It then adds the int values to the heap and prints them out from the heap
Then frees to stack to stop memory leaking

The main problem I faced in this program was I set tit the stack as an array to malloc and I could not get the syntax right a friend looked over and said that I had already set the malloc to be the size of 3 ints so I did not the the array of 3 to assign values too once I changed the int stack from an array everything worked fine

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

int main()
{
      int* stack = 0 ;

      stack = (int*)malloc(3*sizeof(int));

      if (0 == stack)
      {
            printf("incorrect malloc");
      }

      else
      {
            stack[0] = 150;
            stack[1] = 275;
            stack[2] = 400;

            printf("First element: %d\n", stack[0]);
            printf("Second element: %d\n", stack[1]);
            printf("third element: %d\n", stack[2]);

            free(stack);
```

```
		stack = 0 ;

	}

	return 0;

}
```



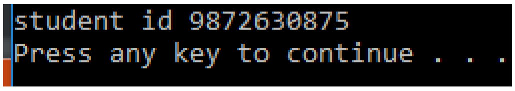
```
First element: 150
Second element: 275
third element: 400
```

Entry 30
Time 4pm 25/05/16
Finish 6pm

Print student it function

In this program I create a function that prints my student id I then call the function from the main

```
#include <stdio.h>

void print_student_id(void)
{
	printf("student id 9872630875\n");
}

int main()
{
	print_student_id();

	return 0;
}
```



```
student id 9872630875
Press any key to continue . . .
```

Memory leak

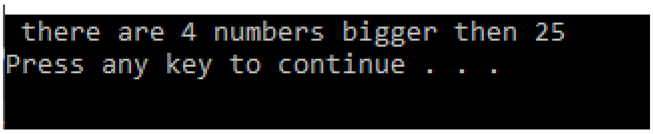
In this program I have to free the memory from the malloc my code became this

```
#include <stdlib.h>

struct Memory_Buffer
{
	unsigned char buffer[64];
};

struct Memory_Buffer* get_memory_block()
{
	return (malloc(sizeof(struct Memory_Buffer)));
}

int main()
{
	struct Memory_Buffer* p_buffer = 0;
	p_buffer = get_memory_block();

	for (int k = 0; k < 64; ++k)
	{
		p_buffer->buffer[k] = k;
	}

	free (p_buffer);
	return 0;
}
```

Simple array, with selection

In this program we go thought the array and find the numbers greater then 25 then print them out to the screen

```
#include <stdio.h>

int main()
{
	int data[] = { 12, 28, 33, 18, 27, 10, 13, 91, 15 };
	int counter = 0;

	for (int y = 0; y < 9; ++y)
	{
		if (data[y]>25)
		{
			counter++;
		}
	}

	printf(" there are %d numbers bigger then 25\n", counter);

	return 0;
}
```

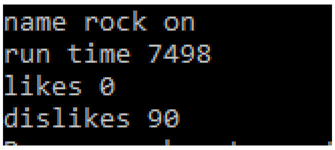


```
 there are 4 numbers bigger then 25
Press any key to continue . . .
```

Passing a song structure, by value:

In this program I create a struct called p1_theme_music I set up some values and in the main function I create a p1_theme_music called song and send corresponding values to that structure I then call that from the main and print it out using another function that prints the structure sent to it.

```
#include <stdio.h>

struct p1_theme_music
{
	char name[30];
	int runtime;
	int likes;
	int dislikes;
};

void print_song(struct p1_theme_music song)
{
	printf("name %s\n",song.name);
	printf("run time %d\n",song.runtime);
	printf("likes %d\n",song.likes);
	printf("dislikes %d\n",song.dislikes);
}

int main()
{
	struct p1_theme_music song;

	sprintf(song.name, "rock on");
	song.runtime = 7498;
	song.likes = 0;
	song.dislikes = 90;

	print_song(song);

	return 0;
}
```



```
name rock on
run time 7498
likes 0
dislikes 90
```

Entry 31
Time 8pm 25/05/16
Finish 10 pm

User defined array size

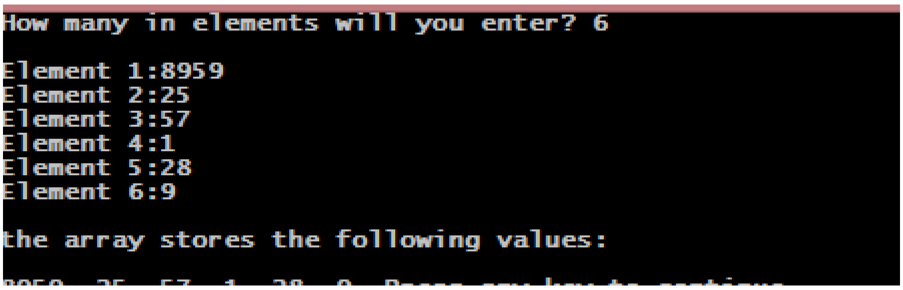
In this program it asked the user to enter a number it then sets up a malloc array to that size I then uses a for loop to ask the user to enter the numbers they want into the corresponding element after that it prints them on the screen

The main problem I had in this one was the syntax for the setting up of the malloc array took me a few goes to get the right way around

code below

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

int main()
{
	int num = 0;
	int count = 1;
```

```
    int* array_num = 0;

    printf("How many in elements will you enter? ");
    scanf("%d", &num);

    array_num = (int*)malloc(sizeof(int)*num);

    printf("\n");

    for (int y = 0; y < num; ++y)
    {
        printf("Element %d:", count);
        scanf("%d", &array_num[y]);
        ++count;
    }

    printf("\nthe array stores the following values:\n\n");

    for (int d = 0; d < num; ++d)
    {
        printf("%d, ", array_num[d]);
    }

    free(array_num);
    array_num = 0;

    return 0;
}
```



```
How many in elements will you enter? 6

Element 1:8959
Element 2:25
Element 3:57
Element 4:1
Element 5:28
Element 6:9

the array stores the following values:

8959, 25, 57, 1, 28, 9, Press any key to continue
```

Entry 32
Time 10.30 am 26/05/16\
Finish

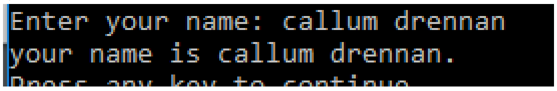
Broken c string

In this program the c string is not correctly set up to take in a sting using malloc I set it up to take in a string of 50 char characters and then print them out from the heap memory

I had a lot of trouble with the syntax of the char malloc origonaly I set it up the same as I would set up an int but I was only setting it to one char ie input = (char*)malloc(sizeof(char)); this would crash my program when I needed to call it
After some tuition from a college I found that I was not setting the size correctly I then changed it to
Input= (char*)malloc(sizeof(char)*50);

My program became this

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

int main()
{
        char* input = 0;

        input = (char*)malloc(sizeof(char)*50);

        printf("Enter your name: ");

        scanf("%[a-zA-Z ]", input);

        printf("your name is %s.\n", input);

        free(input);
        input = 0;

        return 0;
}
```



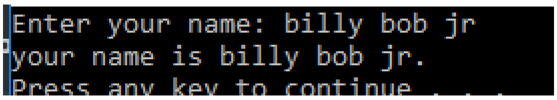
```
Enter your name: callum drennan
your name is callum drennan.
Press any key to continue
```

Another broken c sting

In this program we have to detect where there c sting is wrong and correct it I found that the (char*)malloc(strlen(input)); would not correctly allocate the memory so I changed it to (char*)malloc(sizeof(char)*strlen(input)); and then I freed the memory to stop the leak

Code became this

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

int main()
{
        char input[256];

        printf("Enter you name: ");

        scanf("%255s", input);

        char* p_heap_buffer = (char*)malloc(sizeof(char)*strlen(input));

        strcpy(p_heap_buffer, input);

        printf("Your name is %s.\n", p_heap_buffer);

        free(p_heap_buffer);
        p_heap_buffer = 0;

        return 0;

}
```



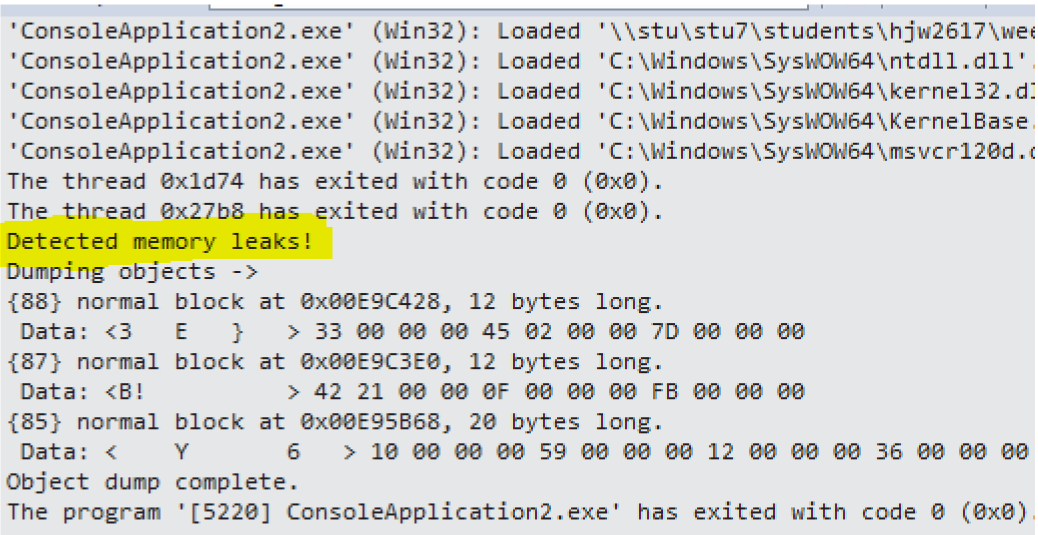
```
Enter your name: billy bob jr
your name is billy bob jr.
Press any key to continue . . .
```

Detect the memory leak

In this program there is a memory leak that has to be found so we include the crtdbg.h and the _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF); command

Program looks like this

```c
#include <stdio.h>
#include <stdlib.h>
#include <crtdbg.h>

int*create_array(int size)
{
      int* p_data = (int*)malloc(size*sizeof(int));

      return p_data;
}

void fill_array(int* p_array, int size)
{
      for (int k = 0; k < size; ++k)
      {
            printf("Enter a value : ");
            scanf("%d", &(p_array[k]));
      }
}

int main()
{
      _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);

      int* first_array = create_array(5);
      fill_array(first_array, 5);

      int* second_array = create_array(3);
      fill_array(second_array, 3);

      int*third_array = create_array(3);
      fill_array(third_array, 3);

      return 0;
}
```

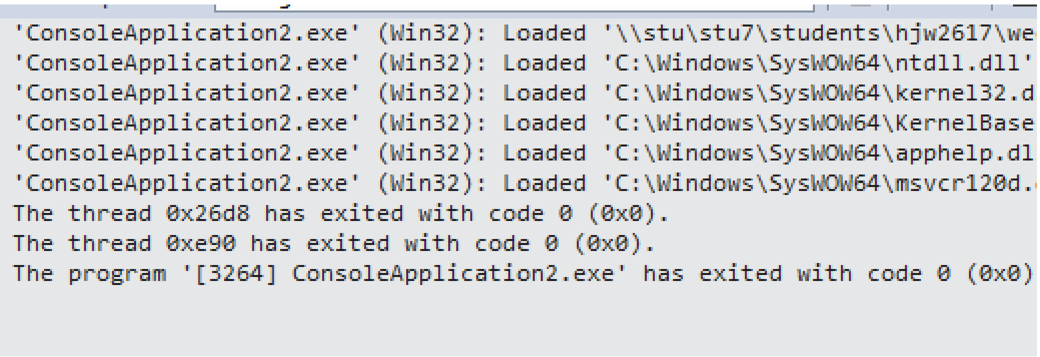


```
'ConsoleApplication2.exe' (Win32): Loaded '\\stu\stu7\students\hjw2617\wee
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ntdll.dll'.
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\kernel32.dl
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\KernelBase.
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\msvcr120d.d
The thread 0x1d74 has exited with code 0 (0x0).
The thread 0x27b8 has exited with code 0 (0x0).
Detected memory leaks!
Dumping objects ->
{88} normal block at 0x00E9C428, 12 bytes long.
 Data: <3   E   }   > 33 00 00 00 45 02 00 00 7D 00 00 00
{87} normal block at 0x00E9C3E0, 12 bytes long.
 Data: <B!          > 42 21 00 00 0F 00 00 00 FB 00 00 00
{85} normal block at 0x00E95B68, 20 bytes long.
 Data: <    Y       6    > 10 00 00 00 59 00 00 00 12 00 00 00 36 00 00 00
Object dump complete.
The program '[5220] ConsoleApplication2.exe' has exited with code 0 (0x0).
```

Fixing memory leaks

This program is the same as before but we are now fixing the problem by freeing the first second and third arrays code below

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

int*create_array(int size)
{
	int* p_data = (int*)malloc(size*sizeof(int));

	return p_data;
}

void fill_array(int* p_array, int size)
{
	for (int k = 0; k < size; ++k)
	{
		printf("Enter a value : ");
		scanf("%d", &(p_array[k]));
	}
}

int main()
{
	_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);

	int* first_array = create_array(5);
	fill_array(first_array, 5);

	int* second_array = create_array(3);
	fill_array(second_array, 3);

	int*third_array = create_array(3);
	fill_array(third_array, 3);
	free(first_array);
	free(second_array);
	free(third_array);
	first_array = 0;
	second_array = 0;
	third_array = 0;

	return 0;
}
```



```
'ConsoleApplication2.exe' (Win32): Loaded '\\stu\stu7\students\hjw2617\we
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ntdll.dll'
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\kernel32.d
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\KernelBase
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\apphelp.dl
'ConsoleApplication2.exe' (Win32): Loaded 'C:\Windows\SysWOW64\msvcr120d.
The thread 0x26d8 has exited with code 0 (0x0).
The thread 0xe90 has exited with code 0 (0x0).
The program '[3264] ConsoleApplication2.exe' has exited with code 0 (0x0)
```

Bibliography

[1] The C++ modulus operator (n.d). retrieved from URL
http://www.cprogramming.com/tutorial/modulus.html

[2] School of Electrical Engineering and Computer Science. (n.d) Input and the scanf function. Retrieved from URL. http://www.eecs.wsu.edu/~cs150/reading/scanf.html.

[3] Allain,A (n.d) Lesson2: If statements in c. Retrieved from URL.
http://www.cprogramming.com/tutorial/c/lesson2.html.

14880673
[4] marco.(2015,april 12).stackoverflow.com[online forum comment]. Message posted to http://stackoverflow.com/questions/10162467/matrix-multiplication-in-c

Hours so far 83.5