

Programming Society:

\$whoami

—

The Committee 2018 - 2019

[aps]

{abertay /*PROGRAMMING*/ society;}

What do I get for my £3?

- Minimum society fee is £1, £3 opens up options to do something meaningful
- Society events
 - Food -> Probably pizza (at said events)
 - Prizes*
- To purchase society hoodies / 'zoodies' - More on that later
- *To keep coming along to society...*
- We are also open to suggestions!

**If budget allows*

Hoodies / Zipped Hoodies Est Cost

£18.50 blank, ~£21 with custom back text for zipped

£16.00 blank, ~£18.50 with custom back text for pullover

In talks to reduce prices by a further 20% (~£17/~15)



Introduction
to
Programming
Society

The Committee:
"Things I wish
I'd known"

< 5 mins

15 mins

30 mins

* mins

Projects we have
done - which
hopefully will inspire
some of you

Discussion



[aps]

```
{abertay /*PROGRAMMING*/ society;}
```

```
def getSkills(atSociety):  
    if atSociety:  
        return "mad"  
    else:  
        return joinSociety()
```

```
Events = {  
    #TODO: add events
```

Data Structures and Algorithms - 2nd:

Shortest path finding algorithms

Multithreaded CPU password cracker, using a dictionary attack.

1

Bodiabaduge 'Gayan' Senaka Jayasekera
- Treasurer

Deception System - 4th - Hons:

An ongoing project in which fake vulnerabilities are used to stall potential attackers, also involves masking OS signatures using packets specially crafted with the scapy library; linux made to appear as a windows system.

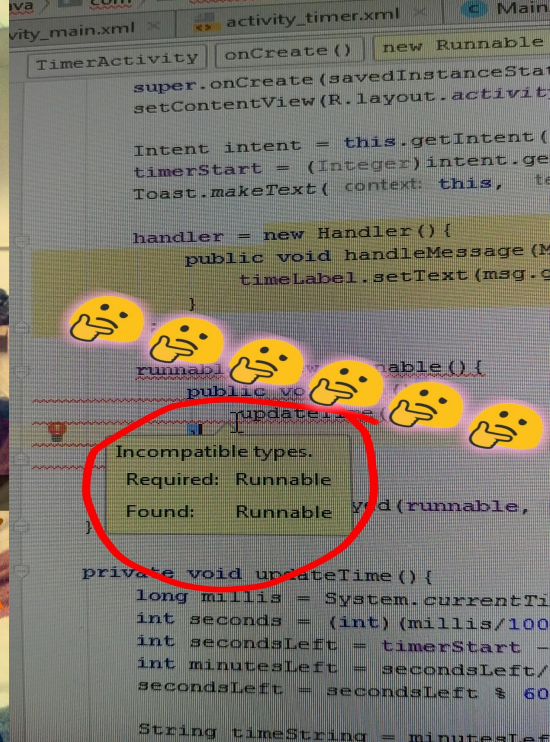
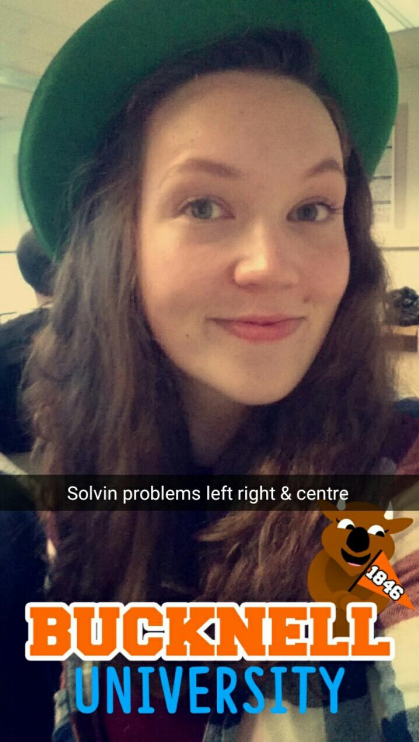
Exploit Development - 3rd:

Scripts to work out the required values to exploit an mps player.

Simple irc botnet.

Android Development - 3rd:

Built a game with a connection back to a server to perform malicious activities.



2

Jessica Sammon
- Secretary

3rd year : kitchen keeper android app

App linked to remote databases for you to keep track of what you've already bought (don't talk to me about android)

2nd year : data mining with R

Program that does sentiment analysis against tweets from 3 biggest cities per state to get most happy & sad states in the US

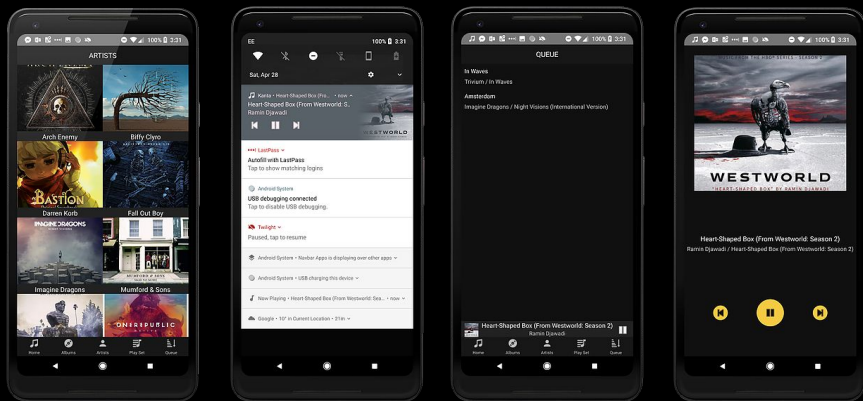
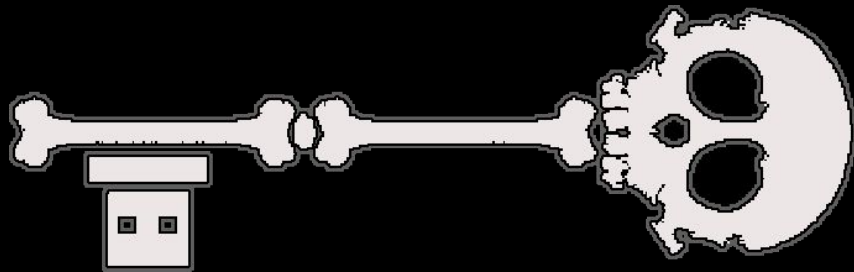
4th year : ?

Ask me again in ~ 2 months, haha

2nd year : team programming competitions

Mainly hackerrank style questions

- Code:PSU, Bucknell, Penn College



Highly Efficient Parallel Packet Processing for Fibre Network Intrusion Detection Systems - 4th Year - Hons
Signature based NIDS with as many CUDA optimisations as I can complete; packet pre-filtering, buffered loading, GPUDirect (GPU speaks directly to NIC), persistent kernels, and all that fun stuff!
CUDA/C/C++

Private until submission

3

Andrew Calder
- *President*

USB Skeleton Key - 3rd Year - Group Project

A RPi Zero W based modular USB Penetration Testing tool, written in Python 3.

<https://github.com/AR-Calder/usbskeletonkey>

<https://www.youtube.com/watch?v=-GGDomc0sIQ>

Kanta - 3rd Year - Android Dev

A music player app that utilized a number of Android 8.0 features, written in Java.

<https://github.com/AR-Calder/Kanta>

<https://photos.app.goo.gl/qjV4bKiGrdCduqxcA>

PassRecovery - 2nd Year - Data Structures & Algo

A MD5 multithreaded brute force password cracker, written in C++.

<https://github.com/AR-Calder/PassRecovery>

Sorry for lack of good readme

Week 0:

Things I wish I'd known

—

The Committee 2018 - 2019

[aps]

{abertay /*PROGRAMMING*/ society;}

Things I wish I'd known: ! PROGRAMMING

1 | Humility
- *Gayan*

There is someone better

- Don't compare yourself
- Ask questions (Google first)

Don't be cocky

- Be nice

2 | Get involved!
- *Jessica*

Don't just focus on uni work

- Societies
- Sports teams
- Student representation
- Volunteering/work
- Etc etc! :)

3 | Spend time wisely
- *Andrew*

Knowledge > Performance*

Stack Overflow is often *not* your friend

Take breaks - This applies to all!

Things I wish I'd known: PROGRAMMING

1

Input Validation
- Andrew

More of an issue with C++
Python method are...
Interesting

2

Exception Handling
- Gayan

Try and Catch blocks:

Try block

- 'Attempts' code
- Throws exception

Catch block

- Catches exception
- Error

3

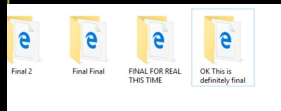
Version Control
- Andrew



I prefer the real version control



I said the *real* version control



Perfection

Input Validation Demo

See Guyans Demo

Exception Handling Demo

```
#include <iostream>

int divide_numbers(int a, int b)
{
    return a/b;
}

int main()
{
    int a, b;
    std::cout << "Number one: ";
    std::cin >> a;
    std::cout << "Number two: ";
    std::cin >> b;

    std::cout << divide_numbers(a, b);

    return 0;
}
```

```
#include <iostream>

int divide_numbers(int a, int b)
{
    try{
        if(b==0)
            //throw an exception
            throw 1;
        return a/b;
    }
    catch(int err_code)
    {
        std::cout << "ERROR CODE " << err_code;
        return err_code;
    }
}

int main()
{
    int a, b;
    std::cout << "Number one: ";
    std::cin >> a;
    std::cout << "Number two: ";
    std::cin >> b;

    std::cout << divide_numbers(a, b) << std::endl;

    return 0;
}
```

```
#include <iostream>

int divide_numbers(int a, int b)
{
    if(b==0)
        //throw an exception
        throw 1;
    return a/b;
}

int main()
{
    int a, b;
    std::cout << "Number one: ";
    std::cin >> a;
    std::cout << "Number two: ";
    std::cin >> b;

    try{
        std::cout << divide_numbers(a, b);
    } catch(int& err_code)
    {
        std::cout << "ERROR CODE " << err_code;
    }

    return 0;
}
```

Version Control -

```
7
8 #include <iostream>
9
10 // An alternative reduction algorithm as provided doesn't seem to work properly
11 Wrapper * reduce_trie(Pattern basic_patterns[], Node trie[]){
12     // last node of last pattern is the highest node index
13     int node_count = basic_patterns[COUNT_PATTERNS - 1].last_related_node_id;
14     int current_trie_reduced_id = node_count - 1;
15
16     // Repurpose last nodes and point to end
17     // & copy basic patterns to reduction pattern type.
18     ReductionPattern patterns[COUNT_PATTERNS];
19
20     for (int p = 0; p < COUNT_PATTERNS; p++){
21
```


```
7
8 #include <iostream>
9
10 +ReductionPattern * get_reduction_pattern(){
11 +     return nullptr;
12 +}
13 +
14 +Wrapper * reduce_trie2(Pattern basic_patterns[], Node trie[], int last_nodes[]){
15 +     // last node of last pattern is the highest node index
16 +     int node_count = basic_patterns[COUNT_PATTERNS - 1].last_related_node_id;
17 +     int current_node_index = node_count - 1;
18 +
19 +     return nullptr;
20 +}
21 +
22 // An alternative reduction algorithm as provided doesn't seem to work properly
23 Wrapper * reduce_trie(Pattern basic_patterns[], Node trie[]){
24     // last node of last pattern is the highest node index
25     int node_count = basic_patterns[COUNT_PATTERNS - 1].last_related_node_id;
26     int current_trie_reduced_id = node_count - 1;
27
28     // Repurpose last nodes and point to end
29     // & copy basic patterns to reduction pattern type.
30     ReductionPattern patterns[COUNT_PATTERNS];
31
32     for (int p = 0; p < COUNT_PATTERNS; p++){
33
```

Version Control – Enable Colours!

arcalder@localhost:~/summer/HEPFAC_GPU/HEPFAC_C

File Edit View Search Terminal Help

```
[arcalder@localhost HEPFAC_C]$ git config --global color.ui true
```



File Edit View Search Terminal Help

```
diff --git a/HEPFAC_C/bitoperation.cu b/HEPFAC_C/bitoperation.cu
index 849a331..4301489 100644
--- a/HEPFAC_C/bitoperation.cu
+++ b/HEPFAC_C/bitoperation.cu
@@ -1,83 +1,93 @@
+#include "bitoperation.h"
+#include "types.h"

- host
- int bitSetsReduced(node r *cur_node, int asciiValue){
-
-     int z;
-     int bitsets=0;
-     //BUG Generator ! (25Nov dans mon JDC)
-     for(z=0; z<asciiValue; z++){
-         if((cur_node->bitmap[z/32] & (1 << (z%32) )) != 0){
-             bitsets++;
-         }
-     }
-     return bitsets;
- }

/*
-Host Bit set function
+Host Bit set functions
*/

- host
- int bitSets(node *cur_node, int asciiValue){
-
-     int z;
-     int bitsets=0;
-     //BUG Generator ! (25Nov dans mon JDC)
-     for(z=0; z<asciiValue; z++){
-         if((cur_node->bitmap[z/32] & (1 << (z%32) )) != 0){
-             bitsets++;
-         }
-     }
-     return bitsets;
- }

+// host // Count bits set in a given reduced node
+int popcount_node(NodeReduced *current_node, int current_char_ascii){
+ int i = 0, // index
+ count = 0; // number of set bits

+ do {
+     if(current_node->bitmap[i/32] & 1 << (i & 31) != 0){
+         // corresponding alphabet section is populated and the index offset is not 0
+         count++;
+     }
+ }
+ while (i < current_char_ascii);
+ }
```



File Edit View Search Terminal Help

```
[arcalder@localhost HEPFAC_C]$ git diff master:./bitoperation.h andrew-dev:./bitoperation.h
diff --git a/HEPFAC_C/bitoperation.h b/HEPFAC_C/bitoperation.h
index 985f155..407ea9e 100644
--- a/HEPFAC_C/bitoperation.h
+++ b/HEPFAC_C/bitoperation.h
@@ -1,25 +1,14 @@
-#ifndef BITOPERATION_H_INCLUDED
-#define BITOPERATION_H_INCLUDED
-
-typedef struct node r{
-    unsigned bitmap[8];
-    unsigned offset;
-}node_r;
-
-typedef struct node t{
-    unsigned bitmap[8];
-    unsigned offset;
-    int final;
-    int prev_node;
-    int node_id;
-}node;
-
-#include "default values.h"
-#include "types.h"
-
-/* Prototypes for the functions */
-int bitSetsReduced(node_r *cur_node, int asciiValue);
-int bitSets(node *cur_node, int asciiValue);
-int popcountCell(node *cur_node, int asciiValue);
-int popcountArray(node *cur_node);
-void clearCurrentNode(node treeArray[(1+(PATTERN_NUMBER * WORD_SIZE))]);
+int popcount_node(NodeReduced *current_node, int current_char_ascii);
+int popcount_node(Node *current_node, int current_char_ascii);
+int popcount_cell(Node *cur_node, int current_char_ascii);
+int popcount_array(Node *cur_node);
+void clear_current_node(Node trie_array[(1+(COUNT_PATTERNS * COUNT_PATTERN_CHARS))]);
-
-#endif
\ No newline at end of file
[arcalder@localhost HEPFAC_C]$
```


File Edit View Search Terminal Help

[arcalder@localhost HEPFAC_C]\$ git diff --stat andrew-dev..master

```
HEPFAC_C/bitoperation.cu      | 122 ++++++-----
HEPFAC_C/bitoperation.h       | 25 +++-
HEPFAC_C/buildreductiontree.cu | 503 ++++++
HEPFAC_C/buildreductiontree.h | 12 +-
HEPFAC_C/buildtree.cu         | 237 ++++++
HEPFAC_C/buildtree.h          | 12 +-
HEPFAC_C/default_values.h     | 20 +---
HEPFAC_C/hepfacac              | Bin 40690 -> 0 bytes
HEPFAC_C/hepfacsearch.cu      | 163 -----
HEPFAC_C/hepfacsearch.h       | 5 -
HEPFAC_C/main.cu              | 1010 ++++++
HEPFAC_C/printverifications.cu | 61 +++++-
HEPFAC_C/printverifications.h | 16 +-
HEPFAC_C/triereduction.cu     | 291 ++++++
HEPFAC_C/triereduction.h      | 8 +-
HEPFAC_C/types.h              | 72 +-----
HEPFAC_C_NoCleanUP/main_andrew.cu | 797 -----
17 files changed, 1488 insertions(+), 1866 deletions(-)
```

[arcalder@localhost HEPFAC_C]\$

Version Control

Things I probably should do...

1 | Pseudo Code

- Helps break down the problem.
- Helps structure code into smaller parts.
- Helps avoid rewrite of entire program.

2 | Comments

- Helps to explain hard code to you and reader.
- When coming back to code means it is easier to work on.

3 | Practices

- Practise makes perfect.
- The more you code the faster you get.

Had absolutely no idea what I was on about?

NEXT WEEK:

GIT / Version Control Workshop

Either bring along an existing project, or use our provided sample!

Talks, workshops and meeting suggestions wanted!

Thanks for listening!

—

See you next week