

C2 – Functions and Control Flow

3.1 Part 1

<pre>Prog1 > C main.c > main() 1 #include <stdio.h> 2 3 4 5 6 7 int main() { 8 unsigned long x = 0.0; 9 10 while(1) { 11 x++; 12 printf("Number: %d\n", x); 13 } 14 return 0; 15 }</pre>	<pre>Number: 6327 Number: 6328 Number: 6329 Number: 6330 Number: 6331 Number: 6332 Number: 6333 Number: 6334 Number: 6335 Number: 6336 Number: 6337 Number: 6338 Number: 6339 Number: 6340 Number: 6341 Number: 6342 Number: 6343 Number: 6344 Number: 6345 Number: 6346 ^C</pre>
---	---

<pre>Prog2 > C main.c > main() 1 #include <stdio.h> 2 #include "math.h" 3 4 #define PI 3.1415 5 6 7 8 int main() { 9 unsigned long x = 0.0; 10 float y; 11 12 while(1) { 13 x++; 14 printf("X: %d", x); 15 16 y = sin(x * (PI / 180)); 17 printf("\tY: %.2f\n", y); 18 } 19 return 0; 20 }</pre>	<pre>X: 28373 Y: -0.93 X: 28374 Y: -0.92 X: 28375 Y: -0.91 X: 28376 Y: -0.91 X: 28377 Y: -0.90 X: 28378 Y: -0.89 X: 28379 Y: -0.88 X: 28380 Y: -0.87 X: 28381 Y: -0.86 X: 28382 Y: -0.86 X: 28383 Y: -0.85 X: 28384 Y: -0.84 X: 28385 Y: -0.83 X: 28386 Y: -0.82 X: 28387 Y: -0.81 X: 28388 Y: -0.80 X: 28389 Y: -0.79 X: 28390 Y: -0.78 X: 28391 Y: -0.76 X: 28392 Y: -0.75 X: 28393 Y: -0.74 X: 28394 Y: -0.73 X: 28395 Y: -0.72 X: 28396 Y: -0.71 X: 28397 Y: -0.69 X: 28398 Y: -0.68 X: 28399 Y: -0.67</pre>
---	---

Prog2 > C main.c > main()

1 `#include <stdio.h>`

2 `#include "math.h"`

3

4 `#define PI 3.1415`

5 `#define FREQ 1`

6

7

8

9 `int main()`

10 `unsigned long x = 0.0;`

11 `float y;`

12

13 `while(1) {`

14 `x++;`

15

16 `y = 0.5 + sin(x * (PI / 180) * FREQ) * 0.5;`

17

18 `printf("X: %d", x);`

19 `printf("\tY: %.2f\n", y);`

20 `}`

21 `return 0;`

22 `}`

X: 27813	Y: 1.00
X: 27814	Y: 1.00
X: 27815	Y: 1.00
X: 27816	Y: 1.00
X: 27817	Y: 1.00
X: 27818	Y: 1.00
X: 27819	Y: 0.99
X: 27820	Y: 0.99
X: 27821	Y: 0.99
X: 27822	Y: 0.99
X: 27823	Y: 0.99
X: 27824	Y: 0.99
X: 27825	Y: 0.98
X: 27826	Y: 0.98
X: 27827	Y: 0.98
X: 27828	Y: 0.98
X: 27829	Y: 0.98
X: 27830	Y: 0.97
X: 27831	Y: 0.97
X: 27832	Y: 0.97
X: 27833	Y: 0.96
X: 27834	Y: 0.96
X: 27835	Y: 0.96
X: 27836	Y: 0.95
X: 27837	Y: 0.95
X: 27838	Y: 0.94
X: 27839	Y: 0.94
X: 27840	Y: 0.94
X: 27841	Y: 0.93

3.2 Part 2

```
void plotval(float val, int width);

int main() {
    unsigned long x = 0.0;
    float y;

    while(1) {
        x++;

        y = 0.5 + sin(x * (PI / 180) * FREQ) * 0.5;

        printf("X: %d\t", x);
        //printf("Y: %.2f\n", y);

        plotval(y, CONSOL_WIDTH);
    }
    return 0;
}

void plotval(float val, int width) {
    char *cons;
    cons = (char *) malloc(width);
    memset(cons, ' ', width);

    int charNum = (int) (width * val);

    cons[charNum] = '*';

    for(int i=0; i<width; i++) {
        printf("%c", cons[i]);
    }
    printf("\n");
}
```

```
X: 1549
X: 1550
X: 1551
X: 1552
X: 1553
X: 1554
X: 1555
X: 1556
X: 1557
X: 1558
X: 1559
X: 1560
X: 1561
X: 1562
X: 1563
X: 1564
X: 1565
X: 1566
X: 1567
X: 1568
X: 1569
X: 1570
X: 1571
X: 1572
X: 1573
X: 1574
X: 1575
X: 1576
X: 1577
X: 1578
X: 1579
X: 1580
X: 1581
X: 1582
X: 1583
X: 1584
X: 1585
X: 1586
X: 1587
X: 1588
X: 1589
X: 1590
X: 1591
X: 1592
X: 1593
X: 1594
X: 1595
X: 1596
X: 1597
X: 1598
X: 1599
X: 1600
X: 1601
X: 1602
X: 1603
X: 1604
X: 1605
X: 1606
X: 1607
X: 1608
X: 1609
X: 1610
```

3.3 Part 3

<pre> #include <stdio.h> #include <stdlib.h> #include "math.h" #include "string.h" #define PI 3.1415 #define FREQ 1 #define CONSOL_WIDTH 80 #define TICK_INTERVAL 10 void plotval(float val, int width, int num); int main() { unsigned long x = 0.0; float y; while(1) { x++; y = 0.5 + sin(x * (PI / 180) * FREQ) * 0.5; printf("X: %d\t", x); //printf("Y: %.2f\n", y); plotval(y, CONSOL_WIDTH, x); } return 0; } </pre>	<pre> X: 540 --- X: 541 X: 542 X: 543 X: 544 X: 545 X: 546 X: 547 X: 548 X: 549 X: 550 --- X: 551 X: 552 X: 553 X: 554 X: 555 X: 556 X: 557 X: 558 X: 559 --- X: 560 X: 561 X: 562 X: 563 X: 564 X: 565 X: 566 X: 567 X: 568 X: 569 --- X: 570 X: 571 X: 572 X: 573 X: 574 X: 575 X: 576 X: 577 X: 578 X: 579 --- X: 580 X: 581 X: 582 X: 583 X: 584 X: 585 X: 586 X: 587 X: 588 X: 589 --- X: 590 X: 591 X: 592 X: 593 X: 594 X: 595 X: 596 X: 597 X: 598 X: 599 --- X: 600 </pre>
<pre> void plotval(float val, int width, int num) { //array of characters to be printed char *cons; cons = (char *) malloc(width); memset(cons, ' ', width); //y axis marker (ish) cons[0] = ' '; //inserts tick into array if num is at tick interval if((num % TICK_INTERVAL) == 0) { for(int i=1; i<4; i++) { cons[i] = '-'; } } //calculates which character should be * int charNum = (int) (width * val); cons[charNum] = '*'; for(int i=0; i<width; i++) { printf("%c",cons[i]); } printf("\n"); } </pre>	

The speed is determined by how many lines of code the computer has to execute between each loop. When there was only printing out the numbers and no complex graphics, the program ran significantly faster. Therefore if you

want your program to run faster, only use the most necessary lines of code to get the job done.

4. Optional Additional Work

In experimenting with changing the colours in the windows command line, I came into a problem in that it would print the change colour command as text. After discussing with my supervisor we came to the conclusion that it was a system error and nothing to do with the program I had written. I did some research into changing the colour of the windows command line and found this Microsoft website which stated that using the command “color (num)” would change the colour off all previous text and would therefore not work on a per character basis. Windows Colour command: <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/color>.

<pre>#include <stdio.h> #define RED "\x1B[31m" #define GRN "\x1B[32m" #define BLU "\x1B[34m" #define RESET "\x1B[0m" int main() { printf(RED "Hello World!" RESET); return 0; }</pre>	<pre>D:\2020+21\ELEC1201 Programming\C2\Additional>gcc nmain.c -o nmain D:\2020+21\ELEC1201 Programming\C2\Additional>nmain.exe ↵[31mHello World!↵[0m D:\2020+21\ELEC1201 Programming\C2\Additional></pre>
---	---

```

#include <stdio.h>
#include <stdlib.h>
#include "math.h"
#include "string.h"

#define PI 3.1415
#define FREQ 1

#define CONSOL_WIDTH 80

#define TICK_INTERVAL 10

void plotval(float val, int width, int num);

int main() {
    unsigned long x = 0.0;
    float y;

    while(1) {
        x++;

        y = 0.5 + sin(x * (PI / 180) * FREQ) * 0.5;

        y = (0.125 + sin(x * (PI / 180) * FREQ) * 0.125);
        y += (0.25 + sin((x * (PI / 80) * FREQ * 2) + (PI / 120)) * 0.25);
        y += (0.25 * (rand() / RAND_MAX));

        printf("X: %d\t", x);
        //printf("Y: %.2f\n", y);

        plotval(y, CONSOL_WIDTH, x);
    }
    return 0;
}

void plotval(float val, int width, int num) {

    //array of characters to be printed
    char *cons;
    cons = (char *) malloc(width);
    memset(cons, ' ', width);

    //y axis marker (ish)
    cons[0] = '|';

    //inserts tick into array if num is at tick interval
    if((num % TICK_INTERVAL) == 0) {

        for(int i=1; i<4; i++) {
            cons[i] = '-';
        }
    }

    //calculates which character should be *
    int charNum = (int) (width * val);
    cons[charNum] = '*';

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

        printf("%c", cons[i]);

    }
    printf("\n");
}

```

```

X: 738
X: 739
X: 740 ---
X: 741
X: 742
X: 743
X: 744
X: 745
X: 746
X: 747
X: 748
X: 749 ---
X: 750
X: 751
X: 752
X: 753
X: 754
X: 755
X: 756
X: 757
X: 758
X: 759 ---
X: 760
X: 761
X: 762
X: 763
X: 764
X: 765
X: 766
X: 767
X: 768
X: 769 ---
X: 770
X: 771
X: 772
X: 773
X: 774
X: 775
X: 776
X: 777
X: 778 ---
X: 779
X: 780
X: 781
X: 782
X: 783
X: 784
X: 785
X: 786
X: 787
X: 788 ---
X: 789
X: 790
X: 791
X: 792
X: 793
X: 794
X: 795
X: 796
X: 797
X: 798 ---
X: 799
X: 800
X: 801
X: 802
X: 803
X: 804
X: 805
X: 806
X: 807 ---
X: 808
X: 809
X: 810
X: 811
X: 812
X: 813
X: 814
X: 815
X: 816
X: 817 ---
X: 818
X: 819
X: 820
X: 821
X: 822
X: 823
X: 824
X: 825
X: 826 ---
X: 827
X: 828
X: 829
X: 830
X: 831 ---
X: 832
X: 833
X: 834
X: 835
X: 836
X: 837

```